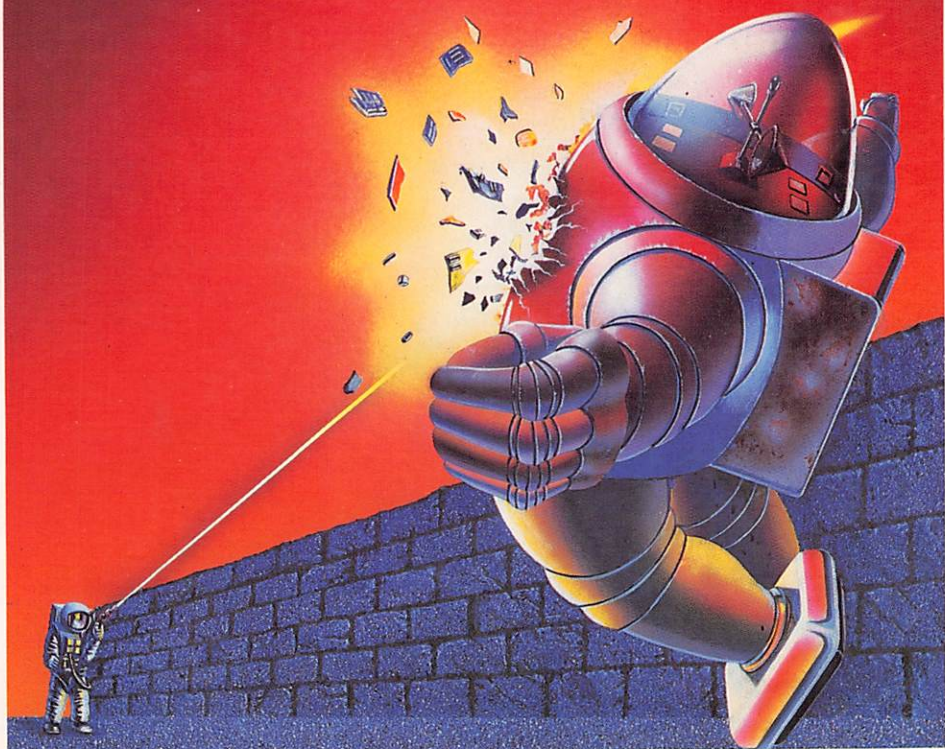


36 Challenging Games

for the **BBC Micro**

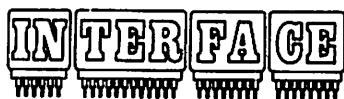
**Tim D Rogers and
Chris Callender**





36 Challenging Games for the BBC Micro

**Tim D Rogers and
Chris Callender**



PUBLICATIONS

44-46 Earls Court Road, London W8 6EJ



First published Interface Publications, 1983
Copyright © Interface Publications, 1983

First printing August 1983

The programs presented in this book have been included for their instructional value. They have been tested with care but they are not guaranteed for any particular purpose. While every care has been taken the publishers cannot be held responsible for running mistakes that may occur.

All rights reserved. No use whatsoever may be made of these programs except for private study by the purchaser of this volume, without the prior written permission of the copyright holder.

CONDITIONS OF SALE

This book is sold subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out or otherwise circulated without the publisher's prior consent in any form of binding or cover other than that in which it is published and without a similar condition including this condition being imposed on the subsequent purchaser.

Any enquiries regarding the contents of this book should be directed by mail to Interface Publications, 44-46 Earls Court Road, London, W8 6EJ.

ISBN 0 907563 37 6

Cover illustration: David John Rowe

Typeset and printed in England by Commercial Colour Press, London E7

CONTENTS

Introduction
A Note about Hi-Scores
Typing in the Programs

PART ONE

Description of Programs

18th Hole; 3D Invaders; Animate; Aspirin; Balloons; Battlefield; Big Business; Breakout; The Clever Computer; 'Copter; Defender; DR 2000; Dodge; Downhill Skiing; Gold Mine; Guess; Hop It!; Invaders; Lander; Morse Test; Nightfall; Parachute; Pilot; Poem; Portals; Rat Attack; Shark; Shock; Slaves; Smash Chamber; Synthesiser; Target; Ups'n' Downs; The Well; Yahm

PART TWO

Displays from Programs

18th Hole; 3D Invaders; Aspirin; Balloons; Battlefield; Big Business; Breakout; The Clever Computer; 'Copter; Defender; DR 2000; Dodge; Downhill Skiing; Gold Mine; Guess; Hop It!; Invaders; Lander; Nightfall; Parachute; Pilot; Rat Attack; Shark; Shock; Slaves; Smash Chamber; Target; Ups'n' Downs; The Well; Yahm

PART THREE

The Program Listings

APPENDIX A: Best Scores

APPENDIX B: Using Joysticks

APPENDIX C: Morse Code

APPENDIX D: Attributions

FOREWORD

Thirty-six outstanding games for the BBC Micro are waiting to be unleashed on your computer.

I've known Tim Rogers for over two years, and in that time have come to admire his remarkable programming skill. He first submitted a number of ZX80 programs to our magazine *Interface*, and I met him at a ZX Microfair. Then, when the ZX81 was released, he provided a number of outstanding programs for the magazine *ZX Computing*.

I was already aware that his programming ability stood head and shoulders above many of the other contributors to the magazine. When Uncle Clive's publishing house *Sinclair Browne* approached me to compile a book of games for the Spectrum (published as *Dynamic Games for the ZX Spectrum*), I asked Tim to contribute. About a third of the book was written by him, proof of his inventive and creative programming skill.

Jeremy Ruston, author of a number of books (including *The BBC Micro Revealed* and *Learn Pascal on your BASIC Micro*) was with me the night Tim Rogers demonstrated the programs he had submitted for the Sinclair Browne Spectrum book. Jeremy commented that the Spectrum programs indicated that, if he had a more sophisticated computer, Tim Rogers could produce even better work. And so we started the discussions which lead to the fine volume you now hold in your hands. From November 1982 to April 1983, Tim worked on the material gathered in this book, consulting frequently with his co-author, Chris Callender.

Like Tim, Chris became known to me via *Interface* and *ZX Computing*. Chris's programs have been included in several books (including *Games ZX Computer Play* and they form the heart of many programs for various computers in the Virgin Books *Games For Your . . . series*) and the programs have consistently demonstrated Chris's determination to produce outstanding programs, which bear the stamp of careful, consistent work. Like Tim, Chris has never been willing to stop working on a program at the point where it first manages to run, bug-free. He knows that the polishing a program receives after this point often makes the difference between 'just another program' and material which is of commercial software standard.

You have here the fruit of many patient hours at the BBC loom; programs which truly deserve the adjective, *Challenging*.

Tim Hartnell,
London, May 1983

Tim Hartnell is the director of Interface Publications. He is the author of some 30 books on microcomputer, including *The Personal Computer Guide* (Virgin Books), *The Book of Listings* (BBC Publications), *Games BBC Computers Play* (Addison-Wesley/Interface) and *Let Your BBC Micro Teach You to Program* (Interface).

Tim D. Rogers lives in the London Borough of Richmond and attends Grey Court Comprehensive School. After doing his 'O'-Levels he hopes to go on to study 'A'-Levels in French, History, Maths and General Studies. His only main hobby is computers, although every now and again he loses a game of squash. He used to ride to school on his pedal-bike, but since moving, he now takes five minutes to walk. He is now sixteen and has to pay full fare on buses and the tube. He has already contributed to various books, and acknowledges the help of his younger brother and sister in testing his games. He is considering a career in personnel management, and his other interests include skiing and politics.

Chris Callender is fourteen years old. He first became interested in computing at the age of 12 years and he has contributed to many publications. Apart from computing, Chris's main interests are sailing and windsurfing. Living at the coast of Scotland's Loch Long, he has every opportunity for water-sports and he spends much of his leisure time on these activities. He is also a keen motorcyclist with an 80cc trials bike, which he rides around the garden at home. A pupil of Glasgow Academy, his favourite subjects are mathematics and science. Eventually, he hopes to combine his aptitude for programming with his ability in science-related subjects.

Introduction

The main purpose of this book is to entertain its owner by providing a source of games which can be typed into a BBC Micro-computer, and enjoyed. We also hope that you will get some inspiration from this volume, or even learn something from the explanation of each program, but you must remember that its prime aim is to entertain.

You will find all sorts of games inside, including the inevitable arcade-type games (which so many love to hate, and others hate to love), but we think that there is something here for everyone.

When writing the programs we had a number of things in mind. The games had to run quickly, and to this aim we have tried to use the resident integer variables as much as possible, and also be in a form which people will find easy to type in, hence the lack of multiple lines; we thought that it would be much easier to type in command after command followed with RETURN rather than a mass of commands separated by colons. Because of this the programs will look much longer than the time it actually takes to type them in.

Throughout the book you will find the authors referring to themselves in the first-person singular, when discussing programs. This is simply to aid understanding and informality. Therefore it is in my role as a single person that I hope you enjoy this book, and the games that it contains.

Tim D. Rogers

LONDON

Chris Callender

HELENSBURGH

Easter, 1983

A Note about Hi-Scores

In some of the programs in this book, the integer-variable H% is used to store the hi-score for a particular game because it is not altered by RUN. Because of this you must type in

H% = 0

before running a game for the first time so that you do not have the hi-score from the previous game interfering.

In some games, the 'highest score' is actually the lowest score, for example in games where you have to do things as quickly as possible. Before playing games of this sort, please enter

H% = 9999

Typing in the Programs

As many of the games in this book are quite lengthy, here are a few words of advice about entering the code.

- * You will make mistakes and so when the program fails to run first time, do not write to complain — check it thoroughly.

- * All the programs in the book run as listed, so before you write to complain borrow a printer and list your version and compare it to the version in this book.

- * Type in exactly what is listed — do not miss out a line because you think it is not necessary, or re-phrase a line because you think it will be quicker to enter.

- * Do not customise the games to suit yourself until you have the program running.

- * You will probably find it easier to enter the programs if a friend reads them to you. If you don't have any friends, record yourself reading the program and play it back.

- * Use AUTO, as all the lines in all the programs go up neatly in tens.

- * Do not enter the ON ERROR lines until the program works.

- * Turn the computer off before typing in a new program, to clear its innards completely.

- * When using AUTO don't forget that you are using it and end up with lines like 5050 PRINT "HELLO".

* Never confuse the letter O or o with the number 0 (zero). We have tried to keep the use of the letter O as a variable down to a minimum.

* Remember not to put spaces in the wrong places. For example, type "PRINT TAB(x,y)" *not* "PRINT TAB (x,y)".

* In MODE 7 it is very easy to confuse brackets '()' with greater or less than signs '<>' (they look almost identical).

* Keep your hair on!

If you finally cannot get a program to run properly after hours and hours, then write to us, but please make it a nice letter — the person who deals with complaints at INTERFACE is very wont to fits of depression.

PART ONE

How to Play

The first section of this book deals with each of the programs, and gives you all the information you need to play each game.

Each program has a description, a 'How It Works' section, information on its length, and a section explaining how to convert the program so that it can run on a BBC Micro Model A.

The 'How It Works' section does not explain in great detail how programs work, but rather highlights points of interest, and gives a brief synopsis of the program structure. Hopefully you will learn something from this.

The 'Length in Bytes' gives the length of the program in bytes, which does not include variable space, or screen space. This is simply to give you some idea of how long it will take you to type it in.

Many of the games run without alteration on both Models A and B. However, all the programs that can only be run on the Model B, have detailed information on how they can be converted to run on a Model A, so that if you have a Model A, you can run *all* the programs in this book.

18th Hole

This game is definitely not for gamers who like playing fast action, arcade-type games. But if you want to show the older members of your family the games-playing capabilities of the BBC Micro, this is the game to show them.

The program prints up a green with a hole near the right-hand edge, and a golf ball somewhere to the left of the screen.

Using your powers of judgement you must try to estimate how hard to hit the ball in order to get it into the hole. You enter the horizontal velocity, and then the vertical velocity. The computer will then move the ball, and will continue to ask you for velocities until you manage to hit the ball in such a manner as to get it into the hole.

My grandparents and their friends particularly liked this game because it does not require finely-tuned reflexes in order to play well. The user can take the game as slow or as fast as he or she likes.

HOW IT WORKS

The program is quite simple in structure. It is run in graphics-mode (by using VDU 5) so that the ball can be moved smoothly (pixel by pixel, rather than character square by character square). Many of the games in this book use VDU 5 so that movement is not jolty, but more realistic.

By using GCOL 3,n (where n is the colour required), movement of objects is made easier than using GCOL 0,n. When GCOL 0,n is used, an object is printed on the screen, but it must be over-printed again in the background colour, so that it appears to move. This means that if you want something red to move across a black background, you have to use GCOL 0,1 (in MODE 2) to print it, and GCOL 0,0 to make it disappear. When GCOL 3,n is used the object is printed in colour n, but

when you want to make the object disappear, you simply print it again using exactly the same statement (GCOL 3,n). In this way you can write a procedure, for example PROCdrawball which draws an object on the screen using GCOL 3,n. When you want the object to be drawn you just say PROCdrawball, and when you want it to disappear you say PROCdrawball again, without having to worry about changing graphic colours, etc. This method of printing objects has the added advantage that if you print the object on another object and then make it disappear, the object underneath will still be there. If you have stars in your display, using GCOL 3,n will mean that they will not start disappearing every time a spaceship moves across them.

STRUCTURE

Line	Effect
10-70	Initialize variables and user-defined graphics, etc.
80-110	Main loop.
120-130	Finish.
PROCIN	Title page.
PROCVA	Sets variables for ball, hole and defines graphics.
PROCBD	Draws golf green with ball and hole.
PROCPB	Prints ball using GCOL 3,n.
PROCHB	Inputs velocities from user and tests for validity.
PROCMB	Moves ball.
PROCGP	Short time loop so that the ball can be seen to move and is not just a blur.
PROCBH	Tells user how many hits it took to get ball into hole.
PROCNG	Asks if user wants another go.
PROCSE	Tests to see if ball is still on the screen.

VARIABLES

B%	x component of ball's position.
H%	x component of hole's position.

T	Time loop variable.
L%	y component of ball's position.
A%	y component of hole's position.
I%	Number of hits so far.
E\$(n)	Reports.
Z%	Horizontal velocity.
V%	Vertical velocity.
D%	Direction of ball.
C%	For/Next Loop.

LENGTH IN BYTES

1540

RUNNING ON A MODEL A

No modifications required.

3D Invaders

In the conventional 'space-invaders' games, the invaders move down towards you, and you can move left and right. In 3D Invaders, you still move left and right, but instead of descending towards you, the invaders come closer. Hence the game requires you to get your laser-bolt in the right vertical plane, and the right horizontal plane (in other 'space-invaders' games you only have to get the bullet in the right vertical plane). Hence, there are two buttons with which you fire: use RETURN to launch a laser-bolt, and when it reaches an invader, press SHIFT to launch the laser-bolt forward towards him/her/it.

The colour of different invaders shows how close they are; cyan and magenta invaders are far away, red and white invaders are very close. Every now and then the whole fleet advances towards you. If a red and white invader gets any closer, it will be in range to fire at you, and the game will end. Use keys 'Z' and 'C' to move left and right respectively, RETURN to launch a laser-bolt upwards, and SHIFT to fire it forwards in mid-air. At the top of the screen is a key showing the various colours of invaders. One is the closest, and eight is the furthest away.

You score points for hitting invaders that are far away.

HOW IT WORKS

The most striking feature about this game is that the invaders are two-coloured.

Usually characters are only one colour. To make a character contain more than one colour you must define two user-defined characters, one for each coloured part of the character. If A\$ is to be the character, you must use various control codes to make it one colour, one character, backspace, another colour, and then another character, and so on. The

backspace means that all the different coloured characters are printed in the same place creating a multi- coloured character.

To make all this clear, type in the following:

```
10 MODE 5
```

```
20 VDU 5
```

Characters of more than one colour will only work in graphics mode (hence VDU 5).

```
30 VDU 23,255,60,102,0,0,0,102,60,0
```

```
40 VDU 23,254,0,0,6,28,6,0,0,0
```

These two characters will make up the final coloured character.

```
50 A$ = CHR.18 + CHR.3 + CHR.1 + CHR.255
```

CHR\$ 18 changes the graphics colour. Thus, CHR\$ 18 followed by CHR\$ 3 and CHR\$ 1 has the effect of GCOL 3,1.

```
60 A$ = A$ + CHR.8
```

CHR\$ 8 is the backspace code.

```
70 A$ = A$ + CHR.18 + CHR.3 + CHR.2 + CHR.254
```

This is the equivalent of GCOL 3,2 with CHR\$ 254.

The character is now complete, and all that is required is to print it:

```
80 MOVE 0,32
```

```
90 PRINT STRING$(10,A$)
```

This will now print a row of ten multi-coloured '3's.

You need not be limited to just two colours. If you have a Model B, you could use a character with 15 colours, although it might look a bit fuzzy.

The program detects a SHIFT button being pressed by using INKEY(- n) where n is the number of the key to be tested. If the key is being pressed the function returns a value of TRUE(- 1), otherwise it returns FALSE(0). In this way you can say:

```
10 IF INKEY( - 1) THEN VDU 7
20 GOTO 10
```

This program will 'beep' every time you press a SHIFT key. Page 275 of the **User Guide** gives the numbers for each of the keys on the keyboard.

The program moves your ship left and right by using INKEY(- 98) for 'Z' and INKEY(- 83) for 'C'. This has the advantage over using INKEY(0) which returns the actual value of the key being pressed at that time, in that it works regardless of whether CAPS LOCK is on or off. Many of the programs in this book use the following logic (or something very similar):

$$X = X + \text{INKEY}(-98) - \text{INKEY}(-83)$$

If X is the x component of the position of the user's spaceship, it will be decremented if he or she presses 'Z', that is, it will be moved to the left, and incremented if he or she presses 'C', moved to the right. Try the following:

```
10 A=50
20 REPEAT
30 PRINT TAB(0,0); A
40 A=A+INKEY(-98)-INKEY(-83)
50 UNTIL INKEY(-99)
```

When you press 'Z' the number in the left-hand corner of the screen will go down and if you press 'C', it will increase. When you press the SPACE-BAR the program will end, because INKEY(- 99) will be TRUE and line 50 will effectively read UNTIL TRUE, in other words, it will not REPEAT the loop any more.

STRUCTURE

Line	Effect
10–80	Initialize the variables and user-defined graphics, etc.
90–210	Main loop.
220–260	Explosions and final trimmings.
PROCIN	Title page.
PROCVA	Initializes variables.
PROCIV	Advances invaders.
PROCBD	Prints up display.
PROCAD	Advances invaders.
PROCIV	Prints invaders.
PROCFI	Launches laser-bolt.
PROCBU	Prints up a laser-bolt.
PROCLL	Rockets laser-bolt at invaders.
PROCMV	Moves user's ship.
PROCEX	Explosion.
PROCNG	Asks user if he or she wants to play again.
1190–1390	Checks to see if an invader has been hit.

LENGTH IN BYTES

2327

RUNNING ON A MODEL A

The program will work if you add the lines below, but because there are only four colours available to you, you will not be able to tell how close the invaders are to you.

```
50 MODE 5
55 VDU 19,2,4,0,0,0
395 IF A%=4 THEN C$(A%)=CHR$18+CHR$0+CHR$1
```

Animate

This program is for those of you who want to produce your own cartoons. By displaying up to 15 frames in quick succession, the program produces convincing animated effects.

When the program is run, the computer displays the words 'Frame 1'. The first frame can be drawn just as in a simple sketch program. The arrow keys are used to draw the sketch. Pressing two keys at the same time produces diagonal lines. When the first frame is completed, pressing the control key, CTRL, will move it on to the second frame. This can be drawn in exactly the same way as the first. Pressing CTRL will move it on to the third frame and so on. When all the required frames have been drawn, they can be displayed quickly, one after the other (to make the cartoon move) by pressing the '@' key. Pressing RETURN will freeze the cartoon. Pressing 'F' will speed up the cartoon and pressing 'S' will slow it down. If the cartoon is speeded up too much, it will not appear to be moving and will look just like a blur.

HOW IT WORKS

It would be extremely slow to draw each frame every time it was needed — far too slow to produce good moving graphics. The command:

`VDU 19,n,m,0,0,0`

can be used to make logical colour 'n' appear as actual colour 'm'. Therefore, if each logical colour is made to appear as actual colour black until it is required and then made actual colour white, the various frames can be made to appear quickly.

LENGTH IN BYTES

755

RUNNING ON A MODEL A

It is not very practical to run this program on a Model A since there are only four colours available on the screen at the same time. This means that only three frames could be used instead of 15. However, if you want to use the program, here are the modifications:

```
60 MODE 5
70 FOR A=1 TO 3
100 FOR F=1 TO 3
```


Aspirin

This is an example of a game which came about almost by accident. I was experimenting with different landscapes, when I got the display for this game. Where most games stem from an idea, it was the other way round with 'Aspirin'.

The idea is that you are an aspirin located at the top of the screen, and you must get to the bottom of the screen through a complex maze of intestine as quickly as possible. You have an infinite choice of stomachs to choose from — just input the number of your choice when prompted at the start of the game. You will then be confronted by various lines and a tiny flashing blip (representing the aspirin) somewhere along the top of the screen. Using a combination of the keys 'Z' (left), 'C' (right), '*' (up) and '?' (down), you must then move diagonally along the intestine and squeeze through to the bottom, where you will be told your time and stomach number.

It can be very tricky at times to move through a stomach (especially at junctions) but if you remember to keep two keys pressed to move diagonally and release one key if you get stuck, you should not find it too difficult. Some stomachs contain junctions through which it is impossible to pass, so it is best to begin playing on 'easy' stomachs, of which number 31 is an example as it contains no impossible junctions.

HOW IT WORKS

The stomach is drawn by randomly using the cursor control codes CHR\$10 and CHR\$11, combined with lines to draw random lines which blot each other out all over the place. The random-number generator is altered according to the stomach number, so that the same stomach is drawn every time the same stomach number is entered.

STRUCTURE

Line	Effect
10-70	Initialize colours, variables, user-defined graphics, random number generator, etc.
80-100	Initialize stomach.
110-170	Move aspirin freely until it enters part of the stomach.
180-330	Main loop.
340-510	Display time, etc.
PROCIN	Title page.
PROCVA	Initializes variables.
PROCUS	User-defined graphics.
PROCMZ	Draws stomach.
PROCPL	Draws lines of stomach.
PROCPP	Displays the aspirin.

LENGTH IN BYTES

1390

RUNNING ON A MODEL A

No modifications required.

Balloons

In this game, you are a blunt tack moving up a wall. As you cannot pop the balloons that confront you, your only hope is to try and avoid them. If you meet a balloon directly, you will be pushed down the screen. The game ends when you are pushed off the bottom of the screen, and hence you score points the longer you survive. Some balloons flash red and cyan, and if you come in contact with these, you not only pass through them freely (instead of being forced down the screen) but you also get a boost and are pushed up the screen, thus lengthening your life as a blunt tack.

At the beginning of each game, the balloons are quite small but as you progress, they grow bigger and soon begin to resemble balloons to a much greater degree.

Use keys "Z" and "C" to move left and right respectively.

HOW IT WORKS

This game uses control codes 30 and 11 (e.g. VDU 30,11 or PRINT CHR\$30;CHR\$11) to move the whole display down one line, hence moving the balloons down. However, so that the blunt tack does not move down with the rest of the display, it is blanked out before VDU 30,11 and then made to reappear, thus effectively remaining static.

STRUCTURE

Line	Effect
10-60	Initialize variables and user-defined characters, etc.
70-80	Reset flash rate.
90-270	Main loop.
PROCVA	Initializes variables.
PROCSH	Prints blunt tack.
PROCEN	Moves tack up the screen if it comes in

contact with flashing balloon but moves it down the screen if the balloon is not flashing.

PROCIN
PROCUS
PROCPC

Title page.
User-defined graphics.
Prints running score.

LENGTH IN BYTES

1234

RUNNING ON A MODEL A

Add or change the following lines:

```
30 MODE 5
35 VDU 19,2,9,0,0,0
220 GCOL 3,1+(RND(2)-1)*2
230 IF RND(100)>90 THEN GCOL 3,2
420 IF POINT (U% + 32, P%) = 2 THEN
P%=P%+32*RND(4):SOUND 0,1,100,10:ENDPROC
510 COLOUR 1
```

Battlefield

This is a game similar to draughts or checkers, except that the board measures nine by eight and captures are made like pawn captures in chess — diagonally forward landing on top of the opponent's piece.

The computer will draw the board and ask you what your move is to be. It will say "FROM?" and you must enter a place on the board from which you want to move. This is done by typing the vertical number and the horizontal number separated by a comma and pressing RETURN (e.g. 5,4). then you must say where you want to move the piece to. This is done in exactly the same way as the 'FROM?' question was answered.

HOW IT WORKS

Line	Effect
10-40	Introduction.
80	User-defined graphics for solid box.
90	Alters colours.
100-190	DATA for board.
200-250	Read DATA into variable B\$.
280-350	Print board.
290-430	Input user's move.
440	Checks that move is not a cheat.
480-540	Call a function, which makes the computer's move.
580-720	Make the computer's move.
740-END	End of game.

LENGTH IN BYTES

1693

RUNNING ON A MODEL A

Make the following modifications:

```
50 MODE 4
90 VDU 19,0,4,0,0,0,19,1,3,0,0,0,
Delete line 260
300 IF S$(A,B)="X" THEN SOUND 1,-15,0,1
310 IF S$(A,B)="O" THEN SOUND 1,-15,200,1
320 IF S$(A,B)<>"X" AND S$(A,B)<>"O" THEN SOUND
1,-15,50,1
```

Big Business

Here is a game for those who would like to make millions of pounds from buying and selling shares, but find that they cannot quite afford the initial investment. There is only one catch and that is that you do not get real money — but considering the amount of money that is usually lost in a game, perhaps that is an advantage!

The computer will draw up a table of the value of the shares in the various companies and tell you how many shares you own in each company. This table will also indicate which function key is associated with each of the companies. Each function key is programmed with the name of the company so that you do not need to type it in. You then input the option that you want. After 30 years, the computer will tell you how much money you have made (or lost!).

HOW IT WORKS

Line	Effect
10–50	Introduction.
250–490	Print table, etc.
500–750	Buy shares.
760–1020	Sell shares.
1030–1190	Plot past record of company on a graph.
1200–1360	Alter value of shares at end of year.
1390–1490	End of program.
1500–1610	Set up definitions of function keys.

LENGTH IN BYTES

3500

RUNNING ON A MODEL A

No modifications required.

Breakout

This is a very fast version of the game where you must hit a ball using a bat at the bottom of the screen, to smash the coloured bricks at the top of the screen. You can, if you are very clever, not only smash through the bricks, but also the back wall. If you can do this award yourself an extra 5,000 points!

'Lightning Breakout' demonstrates the amazing speed of the BBC Microcomputer. On another computer of the same price range, this program would have to have been written in machine code to achieve this speed. If you want, you can slow it down by putting in a delay loop at line 465, but then that would spoil the fun of the game. If you cannot hit the ball as it is, then the delay loop would look something like:

```
465  FOR D=1 TO 50:NEXT
```

The bat is moved using the keys 'Z' (left) and 'C' (right).

HOW IT WORKS

Line	Effect
100-220	Set up the screen.
290	Draws bat.
300	Draws ball.
360-410	Check if the ball has hit anything other than bricks.
440	Moves the bat.
500-580	Check if the ball has hit a brick.
590	Routine when the ball misses the bat.

LENGTH IN BYTES

1186

RUNNING ON A MODEL A

Change line 10 to:

```
10 MODE 5
```


The Clever Computer

Don't you wish that your 'Beeb' could talk to you more genially when you make a mistake in a program? Instead of saying:

No such variable at line 150

it might say:

Sorry, it would seem that you have confused the letter O and number 0 in line number 150, Sir.

Well, this program will not do quite as much as that, but it will enable you to have a fairly intelligent conversation with your computer. If you look at the sample run, you will see what I mean.

The computer will first say:

Hello, how are you today?

You can type in a reply (in capital letters only). The computer will then say something back to you. If, at any stage you want to stop the program, type in:

GOODBYE

HOW IT WORKS

The way that this program works is very interesting. If you type in a sentence such as:

YOUR PRINTER IS GOOD

the computer will look through this until it finds a word that it recognizes. 'PRINTER' is the first word that it recognizes. Then, it chooses one of its responses relating to the word 'PRINTER'. It has three possible responses and one is selected at random. If we were talking about printers, it might say:

Do I even have a printer?

Of course, it does not speak English very well and may sometimes give a stupid response. The INSTR function is used to check for words that it understands in a sentence.

LENGTH IN BYTES

3440

RUNNING ON A MODEL A

No modifications required.

'Copter

In this game you are sitting in a helicopter and must blow up radioactive rods in sequence. You control the helicopter using 'Z' (left) and 'C' (right); pressing '*' will propel the 'copter upwards to combat gravity.

There are ten rods of varying lengths. You must manoeuvre your 'copter to land on the shortest rod, then the next shortest and so on. When you land on a rod it will disappear. The game ends when you land on a rod that is not the shortest (when you blow up), or when you have blown up all the rods.

HOW IT WORKS

Line	Effect
10	Defines 'copter character.
20-60	Title page.
70-110	Input speed difficulty.
120-210	Initialization.
220-480	Main loop.
490-620	Routine when all rods are blown up.
PROCEX	Explosion.
PROCBD	Prints display, including rods.
PROCCP	Prints 'copter.
PROCGP	Delay loop.
PROCEN	Final routine.

LENGTH IN BYTES

1768

RUNNING ON A MODEL A

Make the following alterations:

120 MODE 5

1035 GCOL 0,3

Defender

In this game by Grant Evans from Wales, martians are attacking the planet Earth. To overpower you they must first steal all your fuel. You have been put in charge of fuel defences and must try to stop as many fuel cells as possible being stolen.

Full instructions on how to play are included in the program.

HOW IT WORKS

Line	Effect
170-340	Set up user-defined graphics and envelopes.
350-430	Set initial values of variables.
440-510	Draw aliens on screen.
520-600	Move your ship.
610-740	Draw lasers when they are fired.
750-830	Move aliens.
840-1010	These lines are called when an alien is hit.
1020-1060	Check to see if there are still some aliens left.
1070-1130	Produce explosion sound.
1140-1350	Print instructions.
1360-1420	End of game.
1430	Error handling routine.

LENGTH IN BYTES

4112

RUNNING ON A MODEL A

Change line 90 to:

90 MODE 5

DR 2000

In this great game which Grant Evans has based on a popular horror movie, you must try to kill as many people on a winding, twisting track as you can, whilst trying to avoid the suicide drivers who are coming straight at you.

You steer your car using the usual keys 'Z' and 'C' to move left and right respectively; or if you have joysticks, you can use them instead.

HOW IT WORKS

Line	Effect
140-230	Set up the user-defined graphics for the edge of the road, cars and men.
240-310	Draw the road at position ROADX and width WIDE.
310-330	Print the player's car at position CARX.
350-370	Move the car and produce the engine sound.
420-440	Draw a hitch-hiker.
450-490	Draw the suicide drivers' cars.
500-570	Car hits something
710-750	Ask user whether joysticks are to be used.
760-790	Check where the joystick is.

LENGTH IN BYTES

1735

RUNNING ON A MODEL A

No modifications required

Dodge

This program displays obstacles on the screen, which you must steer around so that you reach the finish in as short a time as possible. You will be asked what difficulty level you want. Level one is very easy and Level ten is practically impossible. After having input the difficulty level, the computer will print obstacles on the screen. When it has finished, it will start the clock and you must try to get the car in the top left-hand corner to the star at the bottom right-hand corner using the following keys:

Z	left
C	right
@	up
/	down

HOW IT WORKS

Line	Effect
40	Asks what difficulty level is required.
100-140	Set up user-defined characters for car pointing in each direction.
160-220	Print obstacles.
330	Prints car, the direction of which depends on the variable DF.
420-450	Move the car.
460-500	Check that the car has not gone off the screen.
510-570	Make sure that the car has not hit an obstacle.
630-760	Draw display, if you crash.
830-930	Tell you the time and how well you did.
940-1000	Ask if you want another go.

LENGTH IN BYTES

1741

RUNNING ON A MODEL A

Alter or add the following lines:

50 MODE 5

80 VDU 19,0,4,0,0,0

280 COLOUR 2

Downhill Skiing

This program will enable you to ski in the comfort of your own home. After all, what is the point of getting soaked and frozen on the side of a mountain somewhere, when you could bring out your BBC Microcomputer, sit down in front of the fire and play this program? Using the keys 'Z' and 'C' to move left and right respectively, you must try to guide your character through the poles, which move up the screen towards it. You must try to make the skier complete the course in as short a time as possible. If your character hits too many poles, the skier will break a leg and will not complete the course. When you finish, the computer will tell you the skier's time to the nearest hundredth of a second, along with the number of poles that the skier hit.

HOW IT WORKS

Line	Effect
60	White background.
70	Makes the screen white.
160	Resets the internal clock.
190-200	Make sound for the program.
240-270	Draw the poles.
280	Puts the skier on the screen.
290	Scrolls the screen.
300	Checks if the skier has hit a pole.
470-580	Calculate the minutes, seconds and hundredths of seconds that the skier took to complete the course and offers the user another go.

PROC__BROKEN__LEG

This is called if you have a broken leg.

LENGTH IN BYTES

1345

RUNNING ON A MODEL A

No modifications required.

1. The first step in running a Model A is to determine the number of units to be produced. This is done by the sales department, which provides a forecast of the number of units to be sold. The production department then determines the number of units to be produced, taking into account the number of units in inventory and the number of units to be shipped.

2. The next step is to determine the number of units to be produced in each of the four quarters. This is done by the production department, which provides a forecast of the number of units to be produced in each quarter. The sales department then provides a forecast of the number of units to be sold in each quarter.

3. The third step is to determine the number of units to be produced in each of the four quarters. This is done by the production department, which provides a forecast of the number of units to be produced in each quarter. The sales department then provides a forecast of the number of units to be sold in each quarter.

4. The fourth step is to determine the number of units to be produced in each of the four quarters. This is done by the production department, which provides a forecast of the number of units to be produced in each quarter. The sales department then provides a forecast of the number of units to be sold in each quarter.

5. The fifth step is to determine the number of units to be produced in each of the four quarters. This is done by the production department, which provides a forecast of the number of units to be produced in each quarter. The sales department then provides a forecast of the number of units to be sold in each quarter.

6. The sixth step is to determine the number of units to be produced in each of the four quarters. This is done by the production department, which provides a forecast of the number of units to be produced in each quarter. The sales department then provides a forecast of the number of units to be sold in each quarter.

7. The seventh step is to determine the number of units to be produced in each of the four quarters. This is done by the production department, which provides a forecast of the number of units to be produced in each quarter. The sales department then provides a forecast of the number of units to be sold in each quarter.

8. The eighth step is to determine the number of units to be produced in each of the four quarters. This is done by the production department, which provides a forecast of the number of units to be produced in each quarter. The sales department then provides a forecast of the number of units to be sold in each quarter.

9. The ninth step is to determine the number of units to be produced in each of the four quarters. This is done by the production department, which provides a forecast of the number of units to be produced in each quarter. The sales department then provides a forecast of the number of units to be sold in each quarter.

10. The tenth step is to determine the number of units to be produced in each of the four quarters. This is done by the production department, which provides a forecast of the number of units to be produced in each quarter. The sales department then provides a forecast of the number of units to be sold in each quarter.

Fruit Machine

This program is a computer version of the popular 'Fruit Machine' game. You must try to make as much money as you can by pressing the 'hold' buttons and stopping the reels.

When you run the program, the three reels of fruit spin on the screen. To stop the left reel, you use the 'Z' key; to stop the centre reel, you use the 'X' key and to stop the right reel, you use the 'C' key.

HOW IT WORKS

Line	Effect
90-240	Set up the graphics characters for the four fruits.
250	Sets up a text window in the middle of the screen, where the fruit will appear.
280-300	Create an array with information on which fruit is at the front of each reel.
400-470	Rotate the reels.
PROC FRUITS	Draws the fruits.
PROCEND	Calculates how much money you have won.

LENGTH IN BYTES

1817

RUNNING ON A MODEL A

Make the following modifications

```
50  MODE 5
80  COLOUR 131
570 IF C=248 THEN COLOUR 3
750 COLOUR 3
```

Gold Mine

This is an adventure program set in Africa. The object of the game is to mine gold.

The function keys 0–6 are used in this program and I suggest that you use a piece of paper like this to show you what each key does.

f0 Move	f1 Lift up	f2 Lift down	f3 Explode	f4 Dig	f5 Finish	f6 Re-start	f7 —	f8 —	f9 —
------------	------------------	--------------------	---------------	-----------	--------------	----------------	---------	---------	---------

First, the computer will draw the mine on the screen. Then it will display an '*'. The computer is now ready to accept a command. Press the required function key and the computer will carry out what you want done.

HOW IT WORKS

Line	Effect
10–50	Introduction.
80–290	Selection of tools.
310	Calls PROCDRAW which draws the mine.
370–490	Main program loop.
380	Draws the man character.
390	Calls FNCOMMAND which asks for a command.
400–460	Call a procedure depending on which command was entered.
470	Erases the man character.
500–600	Draw the mine.
610–650	Draw the sky, sun, clouds and the lift.

670-740	A procedure which draws a box with corners at (X1,Y1), (X2,Y2), (X3,Y3), and (X4,Y4).
750-810	Draw the sky at the top of the screen.
820-950	Draw the sun as a solid circle with rays emerging from it after every 60° of rotation.
960-1120	Draw a cloud at (S,Y) — an oval with bumps on it.
1130-1270	Draw the lift wheel.
1280-1360	Set up the function keys for the game.
1370-1440	A function which waits until a function key is pressed and returns that key.
1450-1490	Draw a man character at position (X,Y).
1500-1520	Erases the man character.
1530-2200	Various procedures which move the man character.
2210-2430	Select a random hazard and check to see if you have the required tools.

LENGTH IN BYTES

4284

RUNNING ON A MODEL A

Make the following alterations:

```

300 MODE 5
305 VDU 19, 3,4,0,0,0
770 GCOL 0,131
830 GCOL 0,2
970 GCOL 0,2
1650 GCOL 0,2
1790 GCOL 0,2

```

Guess

One of the traditional programs which one can almost certainly find in any book claiming to teach beginners how to program is a program in which the user must try to guess the computer's number, aided by hints such as 'higher' or 'lower'.

When I first became very interested in computers, I spent many hours developing a program which would have the computer guessing the user's number. I finally succeeded and the program in this book incorporates two programs, the first where the user must guess the computer's number as quickly as possible, and the second in which the computer must guess the user's. It is fascinating to watch the computer get closer and closer in guesses to the number hidden in your head, and this program easily fulfils the demands of a program which simulates artificial intelligence — it is impossible to tell whether you are communicating with a computer or a person on a terminal somewhere else. The computer is very good at guessing numbers, and usually guesses my number in less than it takes me to guess its number on average.

HOW IT WORKS

The part where you must guess the computer's number is simple enough. The computer guesses your number by using two variables MIN and MAX and it alters these variables according to the hints (higher or lower) which you provide. A random element is incorporated to prevent the program developing an endless loop. A count is used to see how many times the computer is calculating a number, and if it gets too high, the computer decides that its guesses are too cold and alters its range accordingly.

STRUCTURE

Line	Effect
10–100	Main program — calls PROCedures.
PROCIN	Title page

PROCGM	User guesses computer's number.
PROCGY	Computer guesses user's number.
PROCH	Computer's guess is too high.
PROCL	Computer's guess is too low.
PROCS	Computer's guess is fine-tuned.
PROCR	Random element.
PROCW	Displays winner.
PROCHI	Prints 'higher'.
PROCLO	Prints 'lower'.
PROCO	Prints 'correct'.
PROCT	Second title page.
PROCGL	Computer is jubilant.

LENGTH IN BYTES

2164

RUNNING ON A MODEL A

No modifications required.

Hop It!

You are at the bottom of a staircase and boulders are being rolled down from the top to the bottom. By pressing the 'J' key you can jump the boulders which will then roll away safely underneath you. If you jump too soon, or too late, you will be squashed under the debris and you will lose one of your five lives.

HOW IT WORKS

Line	Effect
90	Draws some stairs.
160	Prints a boulder.
170	Prints a man character.
240	Produces sound for the program.
250	Prints the score.
270-310	Make the man character jump.
320-330	Delete the man character and the boulder.
340	Checks to see if the boulder has reached the bottom of the staircase.
360	Checks to see if the boulder has hit the man character

LENGTH IN BYTES

1420

RUNNING ON A MODEL A

Change line 80 to:

80 MODE 5:VDU 23;8202;0;0;0:COLOUR 1

Invaders

In this game by Grant Evans, you must try to repel the alien force and stop them landing on the planet Earth. Only you can save your planet.

The keys 'Z' and 'C' are used to move your base left and right respectively, and the RETURN key is used to fire a laser bolt at the enemy invaders.

HOW IT WORKS

Line	Effect
150-210	Print the aliens.
220-280	Main program loop.
290-350	Wait until a key is pressed and then start a new game.
350-400	Move the invaders down the screen.
410-470	Move your base.
480-620	Fire the laser.
630-700	Updates the score.
710-760	Ask the user how difficult the game is to be

LENGTH IN BYTES

1507

RUNNING ON A MODEL A

No modifications required.

Lander

Most arcade games take place in 'real-time' but do not imitate correctly the laws of physics. In this game you must attempt to land a module onto a slab on a planet's surface. However, it is far more difficult than it sounds because the keys with which you control the module do not directly influence its motion, they merely exert a force on the module. And as every good physicist knows, if you exert a force on an object, it will accelerate. Thus, if you keep your finger on 'Z' to move the module left, it will begin to move but because you are accelerating it, it will rapidly increase in speed. When you release your finger, it will continue to move (why should it stop, there is no other force around to decelerate it). The only way to slow it down is to thrust in the opposite direction (using key 'C'), but keep your finger down too long and it will soon start speeding up in the other direction. The game assumes that the atmosphere will offer some resistance to the motion of the module so it *will* eventually stop, but by that time it might have gone off the screen and out of the range of your scanner.

To further complicate matters, gravity has been taken into account and as this is supposed to be an Earth-like planet, it is set at ten metres per second per second. To combat gravity, use your upward thruster by pressing '*', but do not keep your finger down too long as the module will go shooting off the top of the screen. Remember that the keys affect the acceleration of the module, which in turn affects the velocity; releasing a key will stop the craft accelerating, but it will not stop the craft. The velocity is shown in the top left-hand corner of the screen with arrows showing the direction of the velocity, together with a fuel reading. Fuel is used up when you thrust the rocket left, right or up (gravity is free). When you run out of fuel, you will not be able to thrust anywhere and the module will be left to move under the force of gravity. If you land the module on the slab at more than 300m/s it will explode, as it will if you do not land it squarely on the slab.

This game is quite complicated and requires someone with a delicate touch. Remember not to keep your finger down on a key too long (except for the 'up' key) because the game uses Newton's equations to the letter!

HOW IT WORKS

The game uses six variables for the module, two for the position, two for the velocity and two for the acceleration. The user controls the acceleration, which in turn affect the velocity, which in turn affects the position. The game is run in graphics-mode (VDU 5) because the module is often moved pixel by pixel.

STRUCTURE

Line	Effect
10-60	Initialize variables, etc.
70-90	Draw the display.
100-330	Main loop.
PROCIN	Title page.
PROCVA	Initializes variables.
PROCUS	User-defined graphics.
PROCPS	Prints the ship and velocity.
PROCBD	Displays the planet surface, etc.
PROCEX	Explosion.
PROCGD	Successful landing.
PROCNG	Offers the user another go.
PROCPF	Displays the amount of fuel left.

LENGTH IN BYTES

1755

RUNNING ON A MODEL A

Alter the following lines:

```
50  MODE 5
730 GCOL 0,RND(3)
850 GCOL 0,1
```

Morse Test

This program will help you practice Morse Code. The computer chooses one of its messages, converts it into Morse and plays it through the loudspeaker. It then asks you what it said and you must try to work out what it was. A copy of the Morse Code is in Appendix C.

HOW IT WORKS

Line	Effect
10-50	Introduction.
100-450	DATA to convert English into Morse Code.
490-540	The DATA for the various messages that the computer can play. You can alter these if you want.
570-600	Calls PROC__SOUND.
610-670	Ask what the message was.
PROC__SOUND	Plays a letter in Morse.

LENGTH IN BYTES

2032

RUNNING ON A MODEL A

No modifications required.

Nightfall

You have been hired by the local council to demolish some slum skyscrapers. Armed with a glider and a supply of bombs you set out over the city landscape which will soon be altered. Press RETURN to drop a pair of bombs. Only one pair of bombs may be in the air at one time, so there is no use in hammering away at the fire button — you must, therefore, aim accurately. Because your glider is slowly descending as it sweeps across, you must destroy the taller skyscrapers first to avoid smashing your glider into them and ending the game.

If you succeed in completely flattening the landscape, you will be given a new city with taller skyscrapers and a new glider which goes much faster.

The game ends when you accidentally hit a skyscraper. Points are scored for the amount of the city (or cities) you destroyed.

HOW IT WORKS

Line	Effect
10–60	Initialize variables, etc.
70–210	Main loop.
PROCIN	Title page.
PROCVA	Initializes variables and envelopes, etc.
PROCH	Initializes variables after each sheet.
PROCBD	Draws skyscrapers.
PROCPS	Prints glider.
PROCMS	Moves glider.
PROCEX	Explosion.
PROCB	Drops bomb.
PROCMB	Moves bomb.
PROCPB	Prints bomb.
PROCDB	Destroys the buildings.
PROCS	Displays running score.
PROCS%	Sounds a fanfare.

LENGTH IN BYTES

2207

RUNNING ON A MODEL A

Alter the following lines:

```
30  MODE 5
500 DIM C$(3)
510  FOR A% = 1 TO 3
580  S$ = C$(3) + CHR$253 + CHR$8 + C$(1) + CHR$252
1340 B$ = C$(RND(3)) + CHR$251 + CHR$8 + C$(RND(3))
      + CHR$250
660  PRINTC$(RND(3)) + CHR$255 + CHR$8 + CHR$(RND(4)
      - 1) + CHR$254
```

Parachute

As you fly over a crowded city, your aeroplane develops engine trouble and explodes. Armed with only a parachute, you must try to land without 'splatting' yourself on a building.

When you run the program, you will be asked what the wind strength is. A wind strength of '1' is fairly easy to land in. But if the wind strength is greater than about '5', it is very difficult to land and unless you are very lucky, you will not make it. The program will display the aeroplane flying across the top of the screen. When it is about half-way across, it will explode and a parachutist will emerge from it. Using the keys 'Z' and 'C', to move left and right respectively you must try to get the parachutist onto the ground. The wind will affect the fall making a safe landing very difficult.

HOW IT WORKS

Line	Effect
10-60	Introduction.
70-100	Set up user-defined graphics for buildings, aeroplane, man character and parachute.
140-180	Draw the buildings.
190	Calls the procedure which makes the aeroplane move across the sky.
240	Moves the man character depending on which key is pressed.
300-310	Sound for the program.
400-460	Successful landing.
470-560	Splat!!!!

LENGTH IN BYTES

1142

RUNNING ON A MODEL A

Make the following modifications:

60 MODE 5

130 VDU 19,2,2,0,0,0,19,3,3,0,0,0: COLOUR 2

430 VDU 19,1,12,0,0,0: COLOUR 1

490 VDU 19,1,11,0,0,0: COLOUR 1

Pilot

Have you ever wanted to land your own aeroplane? With this program, you can do just that. . . without having to spend thousands of pounds on a 'plane and flying lessons. What you will need, however, is practice, patience and skill. When you run the program, the computer will display four dials showing the speed and height of the 'plane and its distance and direction from the landing strip. Five keys control the 'plane. These are:

- F — Fast — this increases the speed of the 'plane.
- S — Slow — this slows the 'plane down.
- E — Elevate — this causes the 'plane to climb.
- D — Down — this causes the 'plane to descend.
- C — Change — this alters the direction of the 'plane.

The 'plane is over the landing strip, when the distance dial is reading between 2 and -2. When this happens, you must first bring your height down to zero — when the dial is at the top — and then bring your speed down to zero — by pressing the 'S' key. If you go past the landing strip and your distance needle is at less than -2, you will need to change direction. The 'C' key will do this.

HOW IT WORKS

Line	Effect
80	Calls a procedure, which puts the dials on the screen.
140-170	Flash up the needles on the dials.
180-220	Check if certain keys are being pressed and alter the flight of the 'plane if they are.
230	Sounds the warning alarm.
250	If you miss the landing strip, the computer goes to line 860.
280	Alters the distance of the 'plane from the landing strip.

290–300

Make the engine noise.

310

Calls a procedure, which alters the 'plane's speed and height a little.

780–850

A procedure to draw a circle, which might be of use in your own games. PROC__CIRCLE(X,Y,R) will draw a circle of radius R at (X,Y).

LENGTH IN BYTES

2279

RUNNING ON A MODEL A

No modifications required.

Poem

Many computer poetry programs simply list words in a random order, creating the effect of an abstract poem. This program, however, goes one step further by printing poetry that actually makes sense. Admittedly, the meaning of most of the poem is still abstract, but the structure of the sentences is logical. Some of the lines it comes out with are very funny, and others could have come straight from Wordsworth!

HOW IT WORKS

The data statements are set out logically into nouns, verbs, adjectives, interjections and prepositions. The computer stores the different types of words separately and then uses them to produce sentences. There are eight different sentence constructions which the computer can choose from, and it randomly places verbs and nouns in the appropriate places. The program does not cater for the irregularities of the English language; for example, to form a plural the computer just adds an 's', which is fine for words like 'spider' or 'wood' but not very grammatical when it comes to words like 'sheep' or 'branch'. Similarly, to form a present participle, the computer just adds 'ing', and to form an adverb 'ly', which results in words like 'greenly', 'hateing' or 'swimming'.

However, on the whole, the words are correct and the program must be treated like a two-year-old poetic genius who hasn't quite mastered grammar yet.

STRUCTURE

Line	Effect
10-20	Call title page.
30	Turns on paged mode.
40	Randomizes random number generator.
50-80	Verbs.
90	Prepositions.

100–160	Nouns.
170–200	Adjectives.
210	Interjections.
220–460	Read the words.
470–510	Main loop.
520–950	Sentence constructions.
PROCIN	Title page.

LENGTH IN BYTES

2932

RUNNING ON A MODEL A

No modifications required.

Portals

You are whizzing through space when you come across a time storm, and you are hurled randomly through time. Your only hope is to guide your ship through the portals which rush towards you.

This game is three-dimensional, and as your ship is fixed at the centre of the screen, your surroundings move when you use the keys 'Z' and 'C' to move left and right respectively. When you press 'Z', do not expect the portal to move left as you will move left and, in effect, the portal will actually move right.

You score points for every portal you steer through and as the game progresses, it becomes harder and harder to steer through the portals!

The game ends when you do not go directly through a portal.

HOW IT WORKS

Using GCOL 3,n, the portals are printed and then unprinted as they become larger and larger.

STRUCTURE

Line	Effect
10-70	Initialize variables, etc.
80-190	Main loop.
PROCIN	Title page
PROCVA	Variables and user-defined graphics.
PROCH	Prints ship.
PROCBD	Prints display (not used, but if you want to 'jazz' the program up, use this PROCedure for the initial display).
PROCBG	Draws portal.
PROCNB	Starts a new portal.
PROCBG	Sets how much the ship can move.

PROCEX
PROCH

Explosion.
Offers user another go.

LENGTH IN BYTES
1104

RUNNING ON A MODEL A
No modifications required.

Rat Attack

This game of skill was written by Grant Evans. The object is to save the Earth from a plague of rats by moving your rat hammer over a rat's hole and when the rat comes to the surface, knocking it back down again. The only problem is that there are five holes, but only one person with a hammer. You start with ten lives, and each time that a rat gets out of its holes you lose a life.

To move your rat hammerer left use the 'Z' key and right using the 'C' key. To hit a rat use the '/' key.

HOW IT WORKS

Line	Effect
250-300	Set up user-defined graphics for the rat, the rat hammerer and the ground.
310-350	Draw the ground.
360-420	Move the rat hammerer.
430-450	Print the rat hammerer.
460-550	Move and print the rats.
560-600	If a rat gets out, this makes a sound and moves the rat that got out.
720-740	End of game.

LENGTH IN BYTES

1790

RUNNING ON A MODEL A

Change line 100 to:

```
100 MODE 5:VDU 23;8202;0;0;0;:*FX 11,0
```

Shark

Just when you thought that it was safe to go back into the water . . . In this program, you are a swimmer in the sea, where a shark is circling around you with its jaws wide open. Can you make it to the land?

The sea will be displayed in blue and the land in green. Use the keys, 'Z' to move left, 'C' to move right, '/' to go down and ';' to go up; thus, hopefully avoiding the teeth of the shark.

HOW IT WORKS

Line	Effect
20-50	Introduction.
60-90	User-defined graphics for the sea, the man character and the shark.
180-210	Variable initialisation.
390-420	Move the man character, depending on which key is pressed.
490-510	Called if the man character reaches the land.
620 onwards	If shark eats the man character, these lines are called.

LENGTH IN BYTES

1236

RUNNING ON A MODEL A

Make the following modifications:

```
50 MODE 5
110 VDU 19,2,4,0,0,0,19,3,2,0,0,0: COLOUR 1
150 IF B>2 THEN COLOUR 2 ELSE COLOUR 3
220 COLOUR 130
250 COLOUR 3
280 COLOUR 2
500 COLOUR 3
630 COLOUR 3
```

Shock

In this arcade action game, you use your laser-base to blast aliens which fly down to various columns across the screen. If an alien lands in a column with four aliens already in it, the bottom alien will fall all the way to the bottom and explode. If you are too near the shock-waves of an exploding alien, you will lose a life. You cannot destroy an alien whilst it is moving so it is best to get out of the way when an alien falls to the ground in this manner.

Every now and again a prince-alien appears above one of the three empty wide columns. If you do not destroy it straight away, a bridge of flashing aliens will appear from the left of the screen moving across to the right. You can never destroy any of the aliens in this bridge, and so it is wise to destroy the prince-alien as soon as possible because when the bridge reaches the right of the screen, you lose a life. The bridge stops when you destroy the prince-alien, but starts to build again another as soon as prince-alien appears. On top of all this you will be blasting ordinary aliens, some of whom will be falling on top of you.

You have three lives in this game. Points are scored only for ordinary aliens which are not moving (you score no points for hitting a prince-alien). Use keys 'Z' and 'C' to move left and right respectively, and RETURN to fire.

HOW IT WORKS

Line	Effect
10-90	Call initializing PROCedures, etc.
100-210	Main loop.
PROCIN	Title page.
PROCV	Variables.
PROCBD	Prints up display with columns, etc.
PROCUS	User-defined graphics.
PROCP	Prints laser-base.

PROCSB	Starts laser-bolt.
PROCPB	Prints laser-bolt.
PROCMB	Moves laser-bolt.
PROCMS	Moves laser-base.
PROCSD	Starts alien dropping into column.
PROCPD	Prints alien.
PROCMD	Moves alien dropping into column.
PROCO	Calculates position of column given the column number.
PROCED	Slots alien into column after drop.
PROCPD	Prints running score.
PROCRC	Advances score.
PROCPL	Displays number of lives left.
PROCHT	Laser-bolt has hit something.
PROCSR	Starts alien falling to the ground.
PROCMR	Moves all aliens falling to the ground (may be more than one alien falling at one time).
PROCPR	Prints aliens falling to the ground.
PROCER	Explodes any aliens that have hit the ground and check to see if laser-base was in near vicinity.
PROCMN	Decides whether bridge is to be moved.
PROCSN	Starts prince-alien off.
PROCPN	Prints the bridge.
PROCLN	Moves bridge on.

This program illustrates how much easier it is to write programs using PROCedures. The actual core of the program is only a few lines long, with the subroutines making up most of the data. All the single letter integer variables are used in this game (except for O% to save confusion with zero). Many of the games in this book do one thing at a time: move a ship, shoot a bullet upwards, advance invaders, etc. This program would be useless if this was done in this game since the motion would be incredibly uneven and jolty. Thus, to have each individual part of the game interacting and moving at the same time, various flags are needed to see which parts need attention. For exam-

ple, the flag for the laser-bolt is B%. When a laser-bolt is moving upwards B% takes the value TRUE, so that there is a line IF B% THEN PROCMB, which means 'if the bullet is in the air (B% = TRUE) then go to the procedure which moves the bullet'. When the bullet (or laser-bolt) is not in use, B% takes the value FALSE (=0). Imagine what the program would have looked like without PROCedures! It would simply be a confused mass of spaghetti-coding. When writing programs of your own which you know will be slightly complex, use PROCedures like Vitamin C tablets: you can't have too many of them!

LENGTH IN BYTES

3217

RUNNING ON A MODEL A

Add or alter the following lines:

```
30 MODE 5
32 VDU 19,1,10,0,0,0
35 VDU 19,2,6,0,0,0
37 VDU 19,3,3,0,0,0
740 GCOL 3,3
1560 IF G%<>2 . . . . .(as listed)
2290 GCOL 0,1
```

Slaves

You are aboard a space cargo ship travelling twenty light years from one spaceport to another. You set out with a number of items in your hold, and every light year you come across some other ships who are willing to buy or sell goods. You must remember that when you arrive in a port you will be able to sell your goods at twice as much as the most any space-trader will be willing to offer you. Hence you should buy as many goods as you can, and only sell goods to raise money to buy more valuable goods.

The computer will display what you have and how much money you have, and then tell you how many traders there are for that light year. On pressing the SPACE-BAR, the first trader (if there are any) will either offer you some goods or ask you what you want to buy. After a while you will begin to realise what a good price is for something, but to begin with you will just have to judge using your knowledge of values now. Try as much as possible to sell things for more than you bought them, but when you arrive in port you should have as many goods as possible. When given a choice as to what to buy, you need only type in a few characters of what you want — and using the INSTR function, the computer will then check through to find what you want. If you want coal and type in just 'co', the computer will ask if you want 'coal', and you simply type 'Y' or 'N'. The trader will then give you an offer on coal (actually it is coal dust, but that's neither here nor there).

When you arrive at your destination after twenty light years, the computer will calculate the value of goods in stock and work out your percentage profit (that way, if you start out with £1 million in one game, but only £2,000 in another, you will still be judged fairly).

I shall leave you to figure out yourself why the game is called 'Slaves'.

HOW IT WORKS

Line	Effect
10-120	Main program.
PROCIN	Title page.
PROCVA	Variables.
PROCP	Displays what is in stock.
PROCY	New light year.
PROCBY	Traders.
PROCPR	Calculates price (with half-pences).
PROCDS	Clears screen downwards.
PROCIS	Inputs user's choice.
PROCIB	Displays what is on offer.
PROCM	User has made a mistake.
PROCX	User has made a mistake.
PROCNM	User hasn't enough money.
PROCPR(n)	Returns string of price n (caters for half-pences).
PROCPF(n)	Calculates value of goods in stock.
PROCEN	Arrival in port.
PROCAG	Offers user another go.
PROCCH	Gives user a choice of what to buy or sell.
PROCWT(n)	Delay loop for n/100 seconds.

LENGTH IN BYTES

3728

RUNNING ON A MODEL A

No modifications required.

Smash Chamber

You are speeding around a deserted warehouse with numerous boxes all over the place when you discover that someone has doctored your hover-car so that the brakes do not work and it is constantly accelerating. You can only steer the car around the warehouse, trying your best to miss the boxes. If you turn through more than 180 degrees (reverse), a siren will sound and your car will accelerate even faster. The game lasts as long as you do, and you have four lives.

Steer the hover-car using 'Z' (left) 'C' (right), '*' (up) and '?' (down). If you go off the screen or hit a box your hover-car will explode.

HOW IT WORKS

Line	Effect
10-90	Initialization.
100-230	Main loop.
240-260	End.
PROCIN	Title page.
PROCVA	Initializes variables, envelopes and user-defined graphics.
PROCBO	Displays warehouse.
PROCSH	Starts off hover-car.
PROCPS	Prints hover-car.
PROCD	Moves hover-car.
PROCPC	Displays running score.
PROCNS	New life.
PROCPE	Speeds up hover-car.
PROCEX	Explosion.
PROCDI	Works out car's direction.

LENGTH IN BYTES

2025

RUNNING ON A MODEL A

Alter or add the following lines:

```
60 MODE 5
65 VDU 19,2,4,0,0,0
530 GCOL 0,2
680 GCOL 0,3
760 GCOL 3,RND(3)
890 GCOL 3,1
1000 GCOL 3,1
```

Synthesizer

Using this program, you can produce music on your computer, play the piano or even make the noise of space invaders. You can also enhance the sounds with the special effects channel.

When you run this program, it will ask you which sound effect you want and then you can do as you please.

HOW IT WORKS

Line	Effect
10–120	Ask you which sound you want and display the title.
130	Sets up the required ENVELOPE. These could be replaced by ENVELOPEs of your own or from the BBC User Guide .
160–230	Play sound in the loudspeaker.
260 onwards	A function which returns a value depending on the position of the key pressed.

LENGTH IN BYTES

898

RUNNING ON A MODEL A

No modifications required.

Target

When you run this program, a small 'space invader'-like creature will appear and dash around at the top of the screen. You must try to shoot it down with your laser from your base situated lower down the screen. The program will first ask what speed you want to use the program at. A speed of one is extremely fast, while a speed of 20 is fairly slow.

When you have input the speed that you want, the UFO will appear at the top of the screen. You can move your laser base left or right using the keys 'Z' and 'C' respectively, and you can fire a bullet using the RETURN key. Once you have fired the bullet and it is on its way to the target, the 'Z' and 'C' keys steer the bullet and not the laser base. You have only 30 laser bolts and must attempt to hit the UFO as many times as you can.

HOW IT WORKS

Line	Effect
80-110	Set up user-defined characters for the bullet, the invader and the laser base.
PROCSTAR	Draws stars.
PROCHIT	This is called if you hit the UFO.
PROCMISS	This is called if you miss the UFO.
PROCEND	This is called at the end of the game.

LENGTH IN BYTES

1560

RUNNING ON A MODEL A

No modifications required.

Ups 'n' Downs

This is one of the first 'proper' programs that I wrote on my BBC after spending weeks drawing patterns and playing tunes.

The computer draws 50 squares on the screen and joins some of them with 'ups' and some of them with 'downs'. The 'ups' are green lines and the 'downs' are red lines. It then puts two counters at the start (square 0). Your counter is blue and the computer's is magenta. The computer moves by 'rolling' the dice. It will tell you what it rolled. Then, the message — "PRESS SPACE TO ROLL DICE" — will appear. When you press the SPACE-BAR, the 'dice' will be rolled for you and your counter will be moved. Any counter, which is at the bottom of an 'up' will be moved to the square at the top of the 'up', and any counter at the top of a 'down' will be moved down. The person (or the computer!), who reaches the finish first will win the game!

HOW IT WORKS

Line	Effect
20-30	Set up user-defined graphics for counters.
90-200	Draw the squares on the screen.
210-260	Put numbers in the squares.
280-340	Draw the 'downs'.
350-410	Draw the 'ups'.
430-490	Work out where to put the counters.
510-620	Make the computer's move.
640-810	Make the player's move.
820-830	Check if anyone has won.
860-970	Deal with the computer winning.
980 onwards	Deal with the player winning.

LENGTH IN BYTES

2495

RUNNING ON A MODEL A

The only alteration required is to change line 60 to:

```
60 A=INKEY(100):MODE 5:VDU19,1,4,0,0,0,19,2,5,0,0,0
```

and then, to change all the GCOL 0, 4s to GCOL 0, 1s and all the GCOL 0, 5s to GCOL 0, 2s in lines 430, 490, 610, 790, 870 and 990.

The Well

This is an educational program which can be fun to play even if you do not want to be 'educated'.

The program tests the user's arithmetic by placing him or her in a bucket being lowered into the well of a mad professor. He asks you a question, and if you get it wrong, the bucket gets lower and conversely if you get it right, the bucket gets higher. There are 20 levels — level one tests very basic addition, while level 20 tests slightly more complex sums in all four of the basic functions (addition, subtraction, multiplication and division).

Eventually, the bucket will either be submerged in the water in the well or lifted out of the well completely. If you find yourself submerged in the well, you were obviously on a level that was too high for you. If you get lifted out of the well, the professor is thrown into the water. The quicker you answer questions correctly, the sooner you will get out of the well.

I suggest you start on level 12, and if you find it too difficult, go down a few levels (although I would expect that most adults would be able to cope with level twelve quite easily).

HOW IT WORKS

I tried to make this program as 'user-friendly' as possible, by trapping errors that users may input. For example, on being asked his or her name, if the user enters an empty string, the computer will ask for your name again, and will continue to ask until it gets one, even if it ends up with 'Toe Nail'. The name cannot be more than ten characters long.

STRUCTURE

Line	Effect
10-80	Initialization.
90-180	Main loop.
PROCIN	Title page.
PROCV	Variables.
PROCDW	Draws the well.
PROCBK	Draws the bucket.
PROCTW	Defines the text-window.
PROCD	Inputs difficulty, etc.
PROCDS	Thinks of a question.
PROCQ	Asks the question.
PROCKH	The bucket is submerged.
PROCKP	The professor is thrown into the well.
PROCT(n)	Delay loop for $n/100$ seconds.
PROCNG	Offers user another go.

LENGTH IN BYTES

2652

RUNNING ON A MODEL A

No modifications required.

Yahm

'Yahm' is a game which resembles poker, but is played with dice. After you have thrown five dice, you can re-throw as many or as few of the dice as you wish to get certain combinations. Every go that you have, you must use up one of 13 options. The options are:

Aces, Twos, Threes, Fours, Fives, Sixes, 3 or 4 of a kind, Full House, Low or High Straight, Yahm (five of a kind) and Chance (total of all the dice).

To score Aces, you must throw as many ones as you can, and after three throws, the computer will score for the number of ones you throw. If you throw two ones on the first go, you would obviously leave the ones, and re-throw the other dice, until your three throws have been made. If you end up with three ones, you will score three for your 'Aces' and will not be able to use that option again. If you go for 'Threes', and end up with four threes, you will score 12. The game ends when all your options have been used up. If you have scored 63 or more on your 'upper-half' (Aces, Twos, Threes, Fours, Fives and Sixes), you get a bonus of 35. To get your bonus you will have to get an average of three Aces, three Twos, three Threes, and so on up to three Sixes. The lower half contains three of a kind (where if you throw at least three of the same number the total of all the dice is recorded), four of a kind (the same as three of a kind only you need four), Full House (two of one number, and three of another) where you score 25, Low Straight (a sequence of four numbers, e.g. 2,3,4,5) where you score 30, High Straight (a sequence of five numbers) where you score 40, Yahm (all five dice the same number) which is very difficult to get and you score 50, and Chance in which the total of the dice is scored.

As the game progresses, your options are used up and on your last go, you will only have one choice. If you find you cannot

score any points with a particular set of dice, you will have to choose something for which you will score nought; this is called 'scratching something'.

Because Yahms are so hard to get, you get a bonus of 100 for each extra Yahm you throw during a game, although you will still have to use the set of dice for something else. For example, if you have used your Yahm and thrown five sixes, you will score a bonus of 100, but will have to use the five sixes on something else, like Four of a Kind, or Full House.

The game may sound complicated, but once you get the hang of it you will find it very simple indeed.

HOW IT WORKS

The reason the program is so long is that the computer checks everything you choose. If you choose a Yahm, the computer will check that you have five of a kind, and if not, will score zero.

To shake the dice, the user presses buttons one to five according to the dice he or she wants to throw. If you make a mistake, you may simply press that button again, and the choice will be cancelled. This is done by using an on/off system. If the dice are to be thrown, the variables for each die are 'on'. If an 'on' variable is chosen, it is switched 'off' and *vice versa*.

STRUCTURE

Line	Effect
10-30	Initialization.
40-130	Main loop.
140-200	End.
PROCIN	Title page.
PROCV	Variables.
510-620	Data for options.
PROCSH	Displays scores.

890-980	Check 1-6s.
1000-1100	Check for 3/4 of a kind.
1120-1290	Check for a Full House.
1310-1520	Check for a Straight.
1540-1600	Check for a Yahm.
1620-1640	Calculated for a Chance.
FNT0(n)	Calculates total of dice.
PROCGO	One go.
PROCSC	Scores.
PROCH	Inputs user's option.
PROCPO	Prints points.
PROCLD	Clears certain arrays.
FNST(n\$)	Dotted line into n\$.

LENGTH IN BYTES

3662

RUNNING ON A MODEL A

No modifications required.

PART TWO

Program Examples

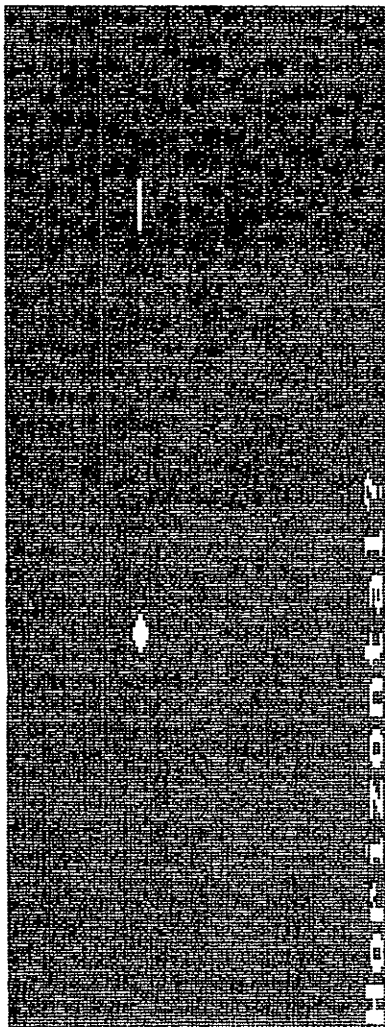
This section of the book is designed to give you some idea of what some of the games look like when you run them. You must bear in mind that they will look much better on your VDU screen, because of colour and movement.

The programs from which displays have been taken are:

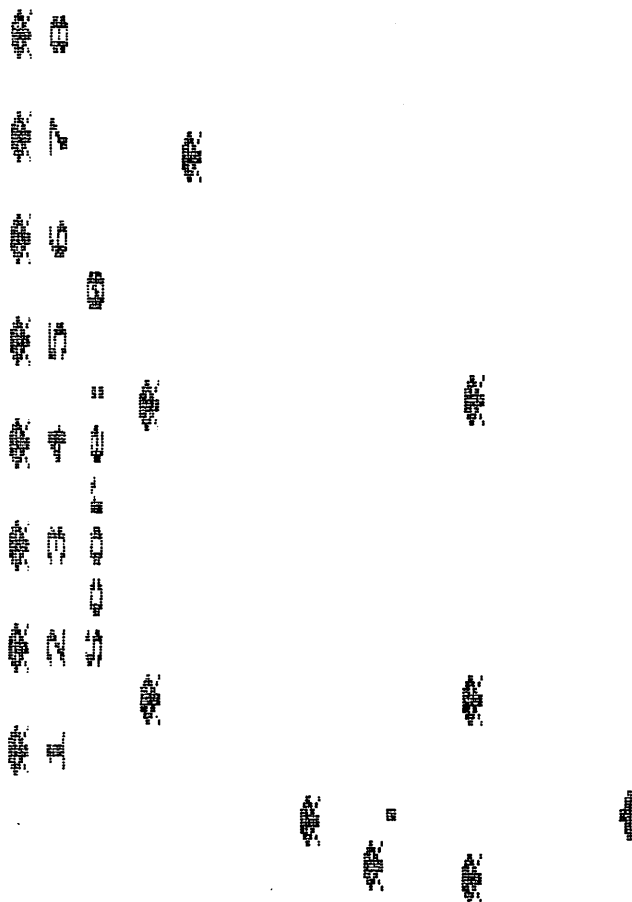
- 18th Hole
- 3D Invaders
- Aspirin
- Balloons
- Battlefield
- Big Business
- Breakout
- The Clever Computer
- 'Copter
- Defender
- DR2000
- Dodge
- Downhill Skiing
- Gold Mine
- Guess
- Hop It!
- Invaders
- Lander
- Nightfall
- Parachute

Pilot
Rat Attack
Shark
Shock
Slaves
Smash Chamber
Target
Ups'n'Downs
The Well
Yahm

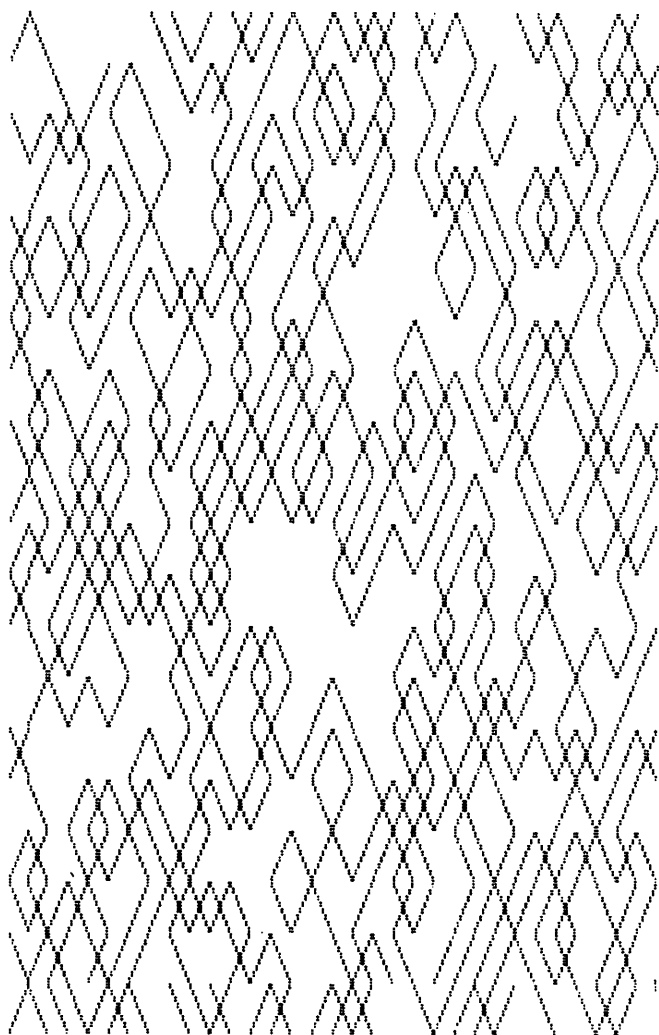
18th Hole



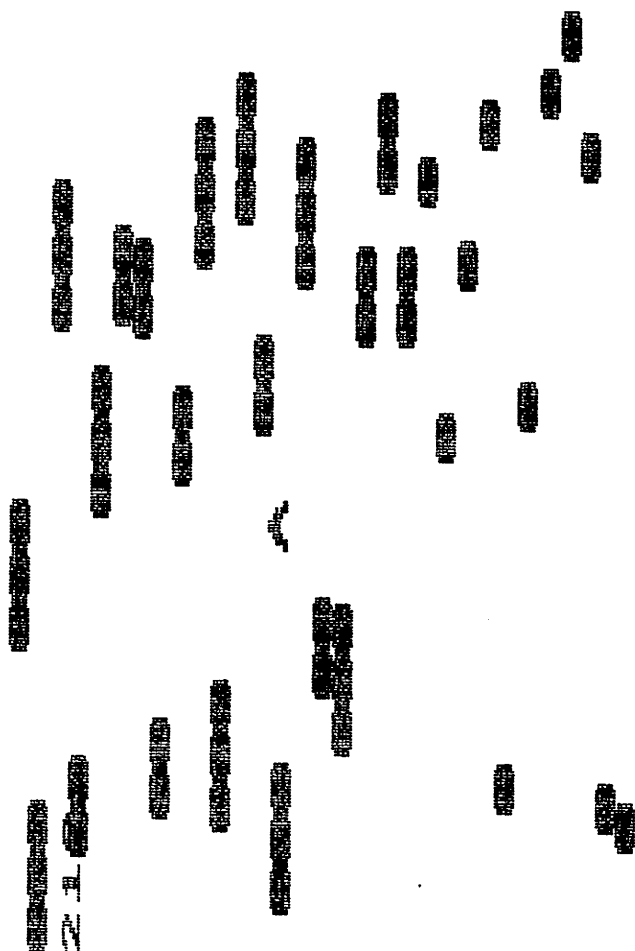
3D Invaders



Aspirin



Balloons



Battlefield

```
#123456789#
1XXXXXXXXXX1
2XXXXXXXXXX2
3XXXXXXXXXX3
4# # # # #4
5# # # # #5
6#0#0#0#0#6
70#0#0#0#07
8#0#0#0#0#8
#123456789#
Your move
From 6,2
To 5,1
```

```
#123456789#
1XXXXXXXXXX1
2XXXXXXXXXX2
3 XXXXXXXX3
4XX# # # #4
50# # # #5
6# #0#0#0#6
70#0#0#0#07
8#0#0#0#0#8
#123456789#
Your move
From 5,1
To 4,2
```

```

#123456789#
1X#X#X#X#X#X1
2#X#X#X#X#X#2
3 # #X#X#X#X3
4#X# # # #4
5 # # # # #5
6# #0#0#0#0#6
70#0#0#0#0#7
8#0#0#0#0#0#8
#123456789#
Your move
From 7,1
To 6,2

```

```

#123456789#
1X#X#X#X#X#X1
2# #X#X#X#X#2
3X# #X#X#X#X3
4#X# # # #4
5 # # # # #5
6#0#0#0#0#0#6
7 #0#0#0#0#7
8#0#0#0#0#0#8
#123456789#
Your move
From 56,54
To 5,3

```

```

#123456789#
1X#X#X#X#X#X1
2# #X#X#X#X#2
3X# #X#X#X#X3
4# # # # #4
5 #X# # # 5
6#0# #0#0#6
7 #0#0#0#07
8#0#0#0#0#8
#123456789#
Your move
From 6,2
To 5,3

```

```

#123456789#
1 #X#X#X#X#X1
2#X#X#X#X#X2
3X# #X#X#X#X3
4# # # # #4
5 #0# # # 5
6# # #0#0#6
7 #0#0#0#07
8#0#0#0#0#8
#123456789#
Your move

```


Big Business

Year 1

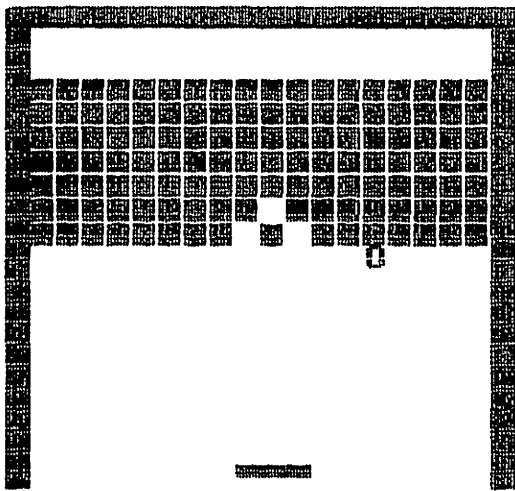
Company	Price	No.	Total
f0: Corna	3.06	0.00	0.00
f1: Ratai	2.97	0.00	0.00
f2: Dargon	2.91	0.00	0.00
f3: Harps	2.81	0.00	0.00
f4: Atndy	2.14	0.00	0.00
f5: Texsa	2.55	0.00	0.00
f6: Inclairs	0.27	0.00	0.00
f7: Commdora	0.23	0.00	0.00
f8: Aplep	1.27	0.00	0.00
f9: Comnas	2.57	0.00	0.00

Money left over `1000.00

Money in shares `0.00

- 1.....Buy Shares
- 2.....Sell Shares
- 3.....Move to next year

Breakout



Score 10
Ball 2

The Clever Computer

Hello, how are you today

? I AM FINE

Your approach to life is very selfish

? COULD YOU EXPLAIN WHY YOU THINK THAT
I am only a computer. I am not human.

? AH. I SEE - COMPUTERS CANNOT EXPLAIN
I am not interested in your silly little
Problems!

? WELL THEN I SEE NO REASON TO CONTINUE
THIS CONVERSATION

Your approach to life is very selfish

? YOU HAVE TOLD ME THAT ALREADY
Do I worry you?

? LET US CHANGE THE SUBJECT
Off a cliff?

? PROBABLY! WHAT ABOUT TALKING ABOUT SOUND
Must I answer that?

? NO, SINCE I HAVE NOT ASKED
I would forget that idea!

? SOUND CAN BE HARD TO PROGRAM
It is loud!

? ONLY IF YOU WERE TO USE SOUND 1,-15,53,1
So what?

? COLOURS IF THAT HELPS
So what?

? COLOUR WAS THE LAST KEYWORD, NOT COMPUTER
Do you like computers that much?

? TALK ABOUT COLOUR COMPUTER
I am a computer

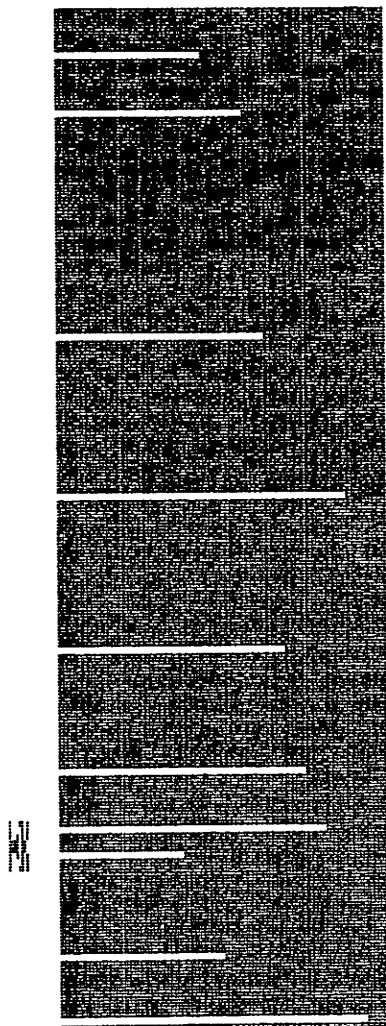
? IF WE WERE TO TALK ABOUT COLOUR THEN
So what?

? YOUR PRINTER IS GOOD
I am glad to hear that!

? THE PRINTER IS PRINTING OUR CONVERSATION
My Printer is sick.

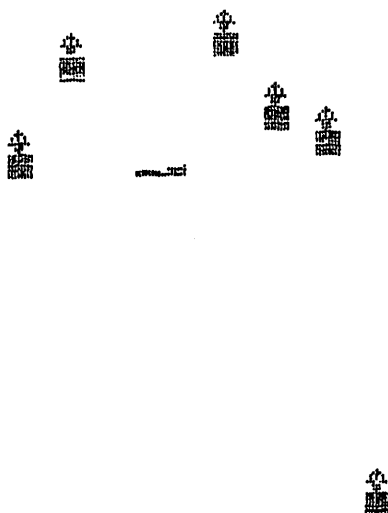
? WELL GOODBYE
Off a cliff?

'Copter

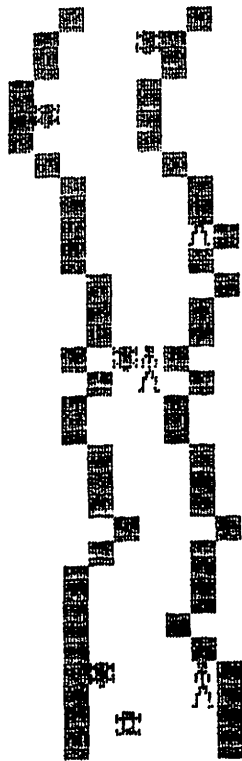


Defender

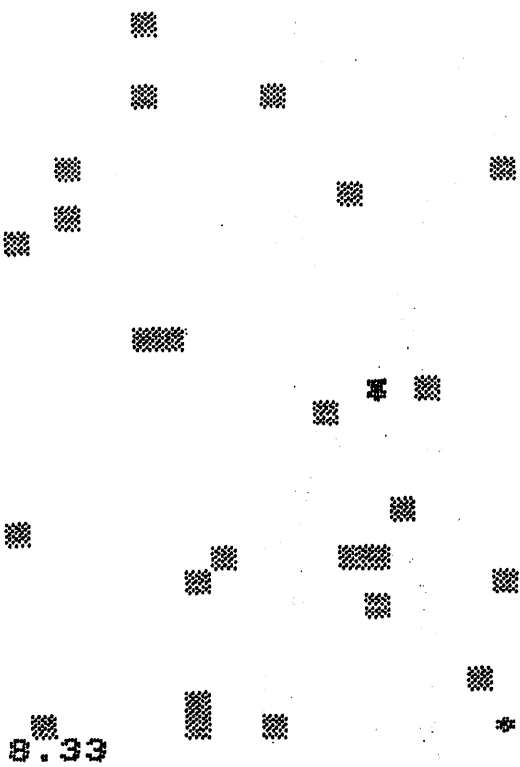
SCORE=250 FUEL=13



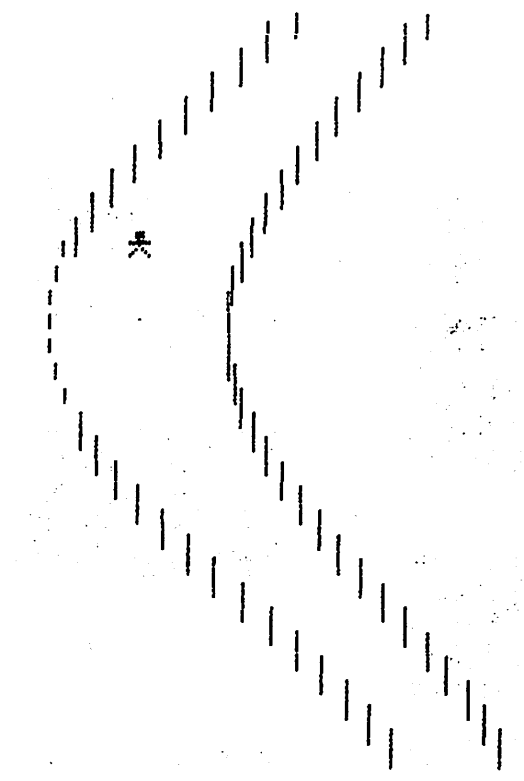
>



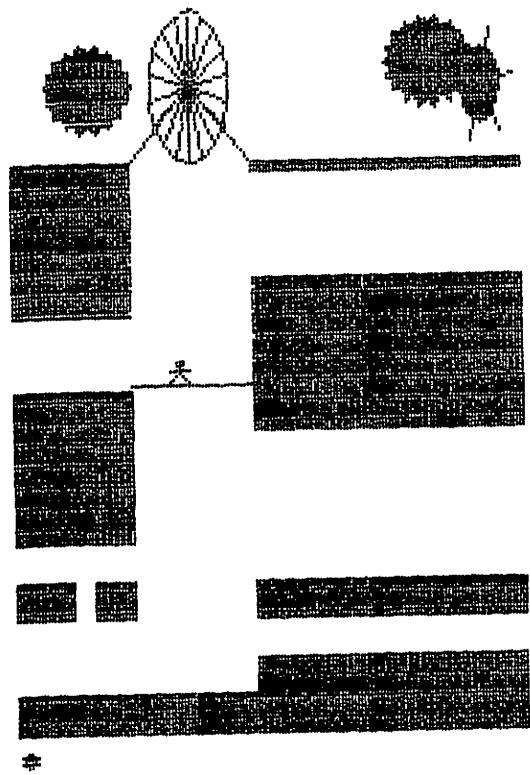
Dodge



Downhill Skiing



Goldmine



Guess

Guess my number...

It is between 1 and 10000

Guess...? 5000

HIGHER

Guess...?7500

HIGHER

Guess...?8300

LOWER

Guess...?7650

LOWER

Guess...?7633

LOWER

Guess...?7580

HIGHER

Guess...?7600

HIGHER

Guess...?7625

LOWER

Guess...?7612

HIGHER

Guess...?7618

LOWER

Guess...?7613

HIGHER

Guess...?7615

HIGHER

Guess...?7617

LOWER

Guess...?7616

Corrrrrrect!

You did it in 15

Just listen to that applause!

Now it's my turn to guess your number

But first you must think of a number

Between 1 and 10000

Hit the space-bar when you're ready

Is it 7319?

LOWER

Is it 3659?

LOWER

Is it 1829?

HIGHER

Is it 2743?

LOWER

Is it 2056?

LOWER

Is it 1949?

HIGHER

Is it 1981?

HIGHER

Is it 2014?

LOWER

Is it 2010?

LOWER

Is it 1995?

HIGHER

Is it 2001?

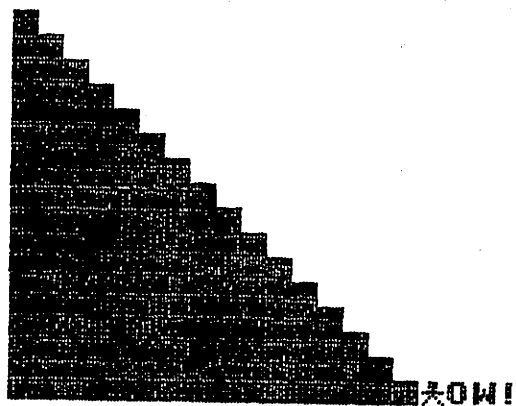
I did it in 11

You did it in 15

But then you're only human...

ANOTHER GO?

Hop It!

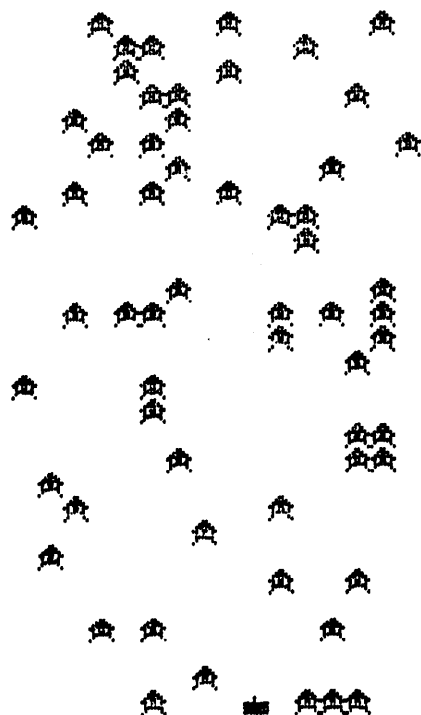


Score 0

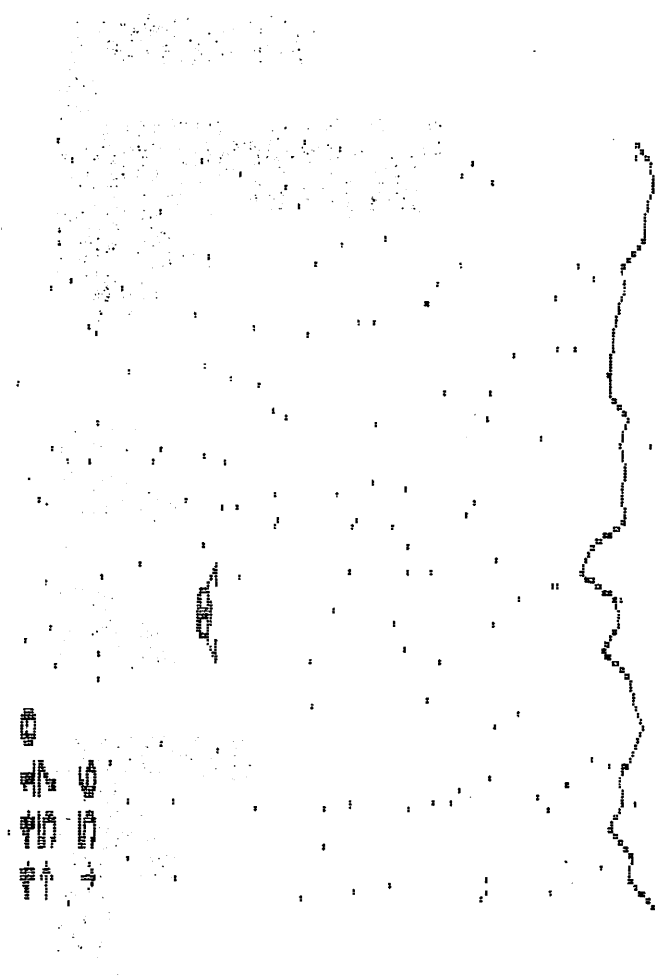


Invaders

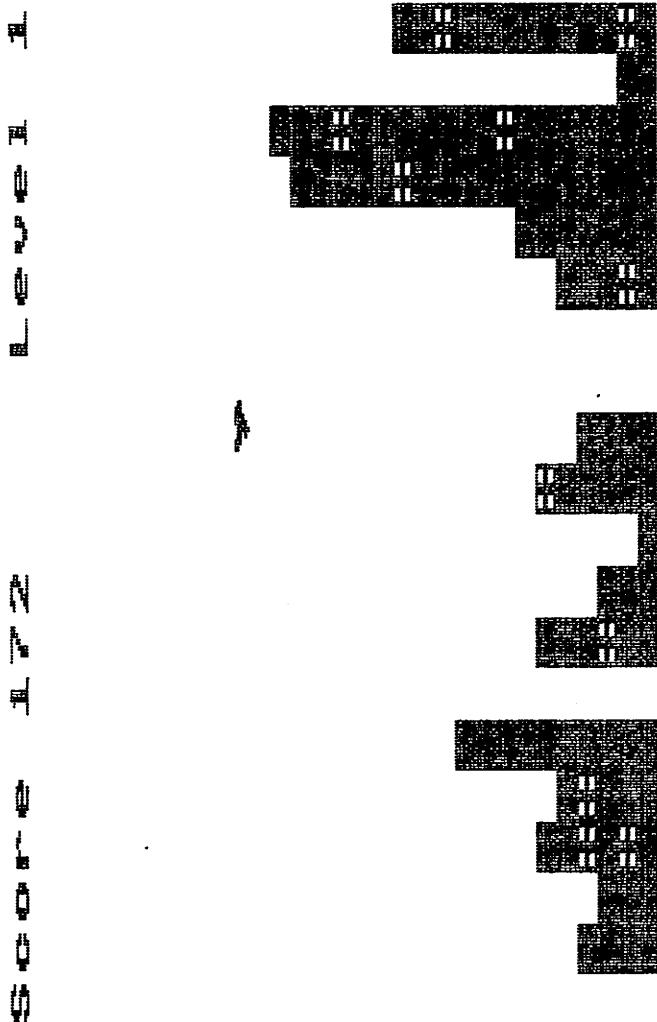
>SCORE-210 HI-220



Lander



Nightfall



Parachute

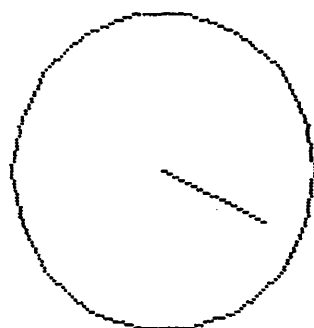
>

5

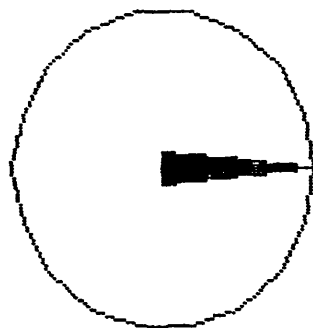


Pilot

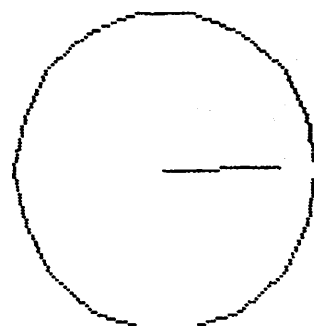
 Tune	 Volume	 Tone	
---	---	---	---



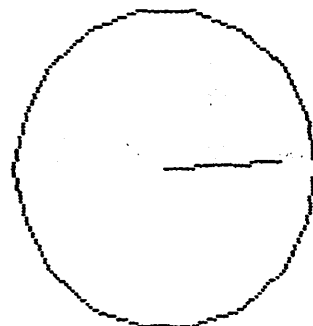
HEIGHT 410



DIRECTION



SPEED 177

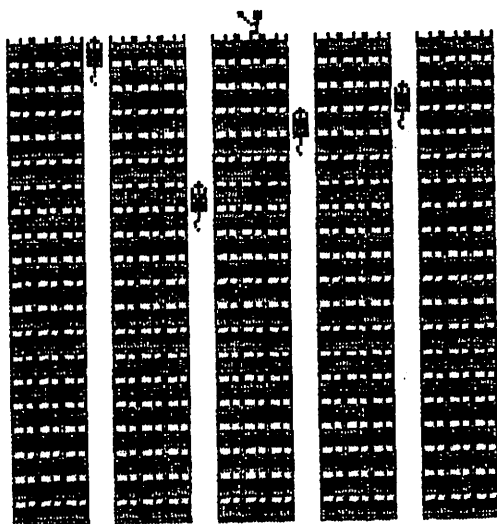


DISTANCE 17

Rat Attack

SCORE-230

LIVES-8

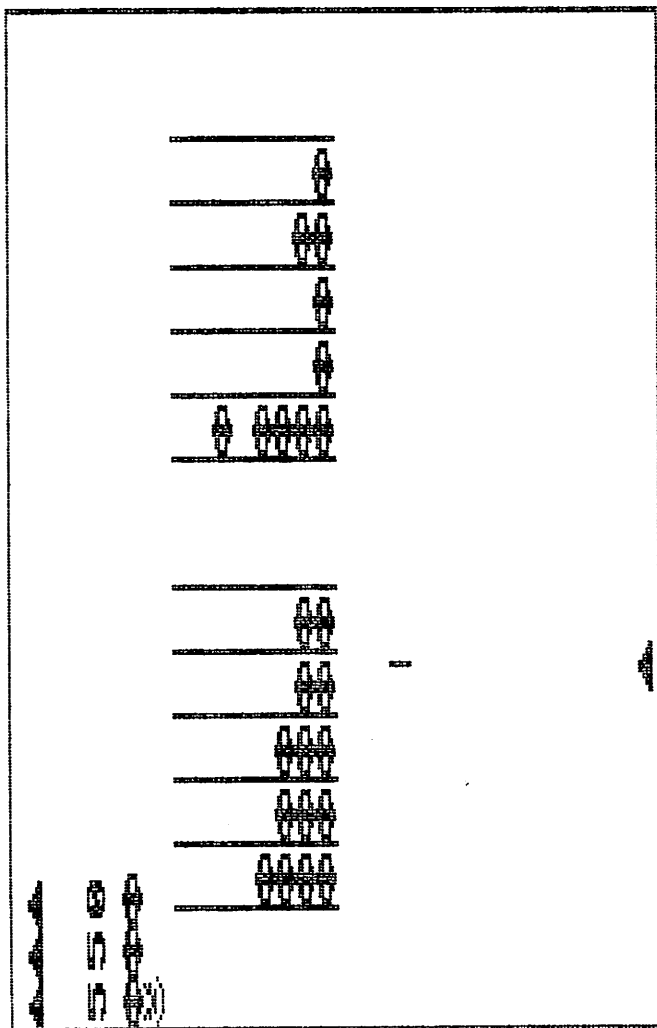


Shark

>



Shock



Slaves

LIGHT YEARS 1

You have:

`16696

1272	pieces of orange peel
8003	tonnes of coal dust
8243	kilograms of grain
1626	test-tubes
9076	slippers
8108	dominoes
2105	search warrants
9583	bars of reject soap
7155	postcards from Milton-Keynes
2762	flower arrangers

There are 4 traders
wishing to sell some goods.

Please hit the space-bar

'What do you want to buy?'

says Trader 1

? smarties

Did you say 'smarties'? Y/N

Trader 1 wants to sell

1897 to 4420

smarties

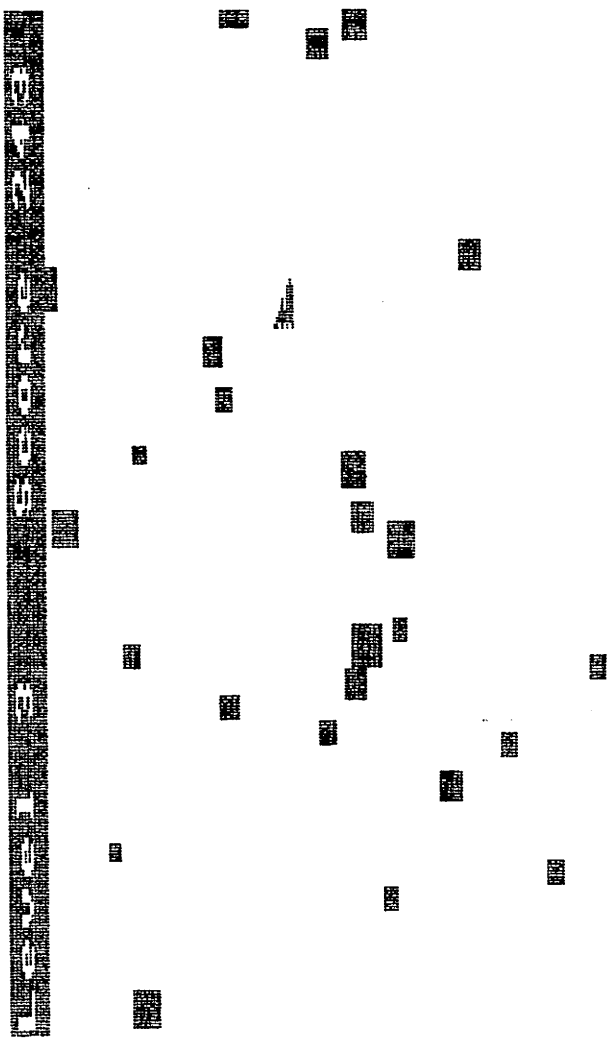
at 23p each

How many to buy?

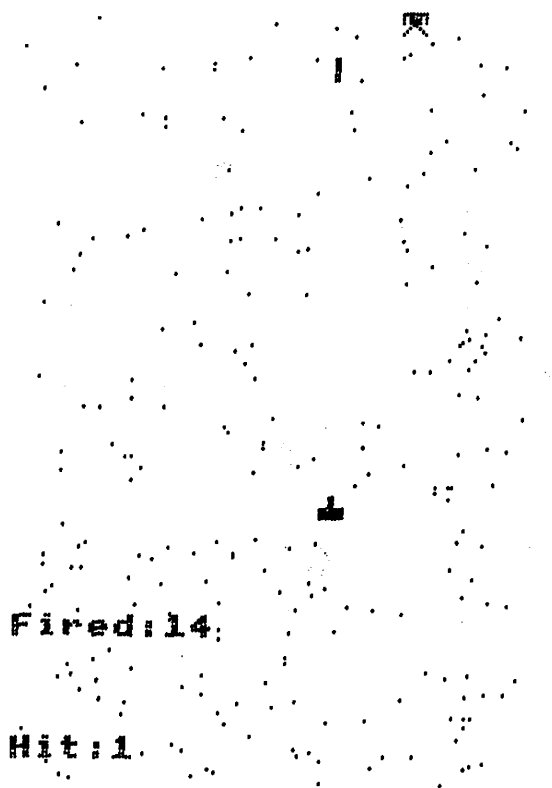
(You have `16696)

?y 3000

Smash Chamber

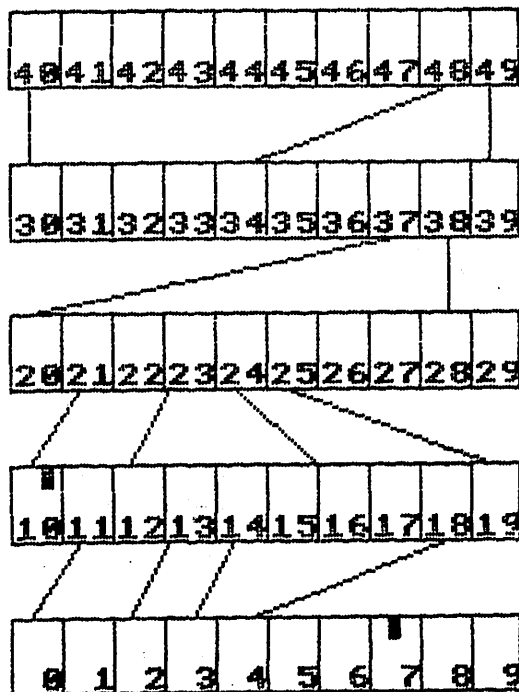


Target

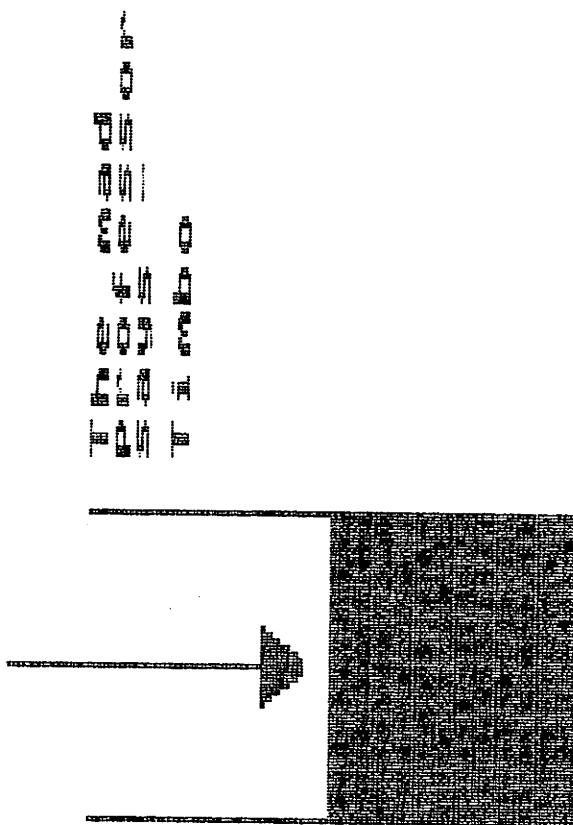


Ups'n'Downs

Ups 'n' Downs
Press SPACE to roll
dice.



The Well



Yahm

YAHM SCORE CARD

Aces.....	3
Twos.....	4
Threes.....	6
Fours.....	0
Fives.....	15
Sixes.....	0
TOTAL.....	28
Bonus (if 63+).....	0
Total of UPPER HALF.....	28
3 of a kind.....	21
4 of a kind.....	23
Full House.....	0
Small Straight.....	30
Long Straight.....	0
YAHM.....	0
Chance.....	22
Total of LOWER HALF.....	96
Total of UPPER HALF.....	28
Grand Total	124
Grand Total	124
hit the space-bar	

PART THREE

The Program Listings

This is the final main section of the book, and contains the actual listings of the programs. The programs have been listed using LISTO7, with a width of forty characters on an Epson FX-80 dot-matrix printer.

If you have any problems reading or understanding the listings, please do not hesitate to write to us at INTERFACE.

Before typing in a program, please read the section on 'Typing in the Programs'.

All the programs were dumped to the printer directly after having been tested, so we will be very surprised indeed to hear of any mistakes in the listings contained in this book.

18TH HOLE

```
0 REM 18TH HOLE
10 CLS
20 MODE 7
30 PROCIN
40 MODE 5
50 PROCVA
60 VDU 5
70 PROCBD
80 REPEAT
90   PROCHB
100  PROCMB
110   UNTIL ABS(B%-H%)<32
120 PROCBH
130 END
140 DEF PROCIN
150 VDU 31,10,10,141
160 PRINT"18th Hole"
170 VDU 31,10,11,141
180 PRINT"18th Hole"
190 T=TIME
200 REPEAT
210   UNTIL INKEY(-99) OR TIME-T>200
220 ENDPROC
230 DEF PROCVA
240 VDU 19,1,2,0,0,0
250 VDU 19,2,6,0,0,0
260 VDU 23,255,0,0,0,255,255,0,0,0
270 VDU 23,254,0,24,60,126,126,60,24,0
280 H%=1000
290 L%=400
300 B%=RND(500)
310 A%=400
320 I%=0
330 DIM E$(3)
340 FOR N%=1 TO 3
350   READ E$(N%)
```

```

360 NEXT
370 DATA You hit the ball      into the
sea!!!!
380 DATA You knocked the clubhouse chi
mney off!!
390 DATA You smashed a green house!!
400 ENDPROC
410 DEF PROCBD
420 GCOL 0,1
430 MOVE 1279,0
440 PLOT 85,1279,600
450 MOVE 0,0
460 PLOT 85,0,600
470 GCOL 0,2
480 MOVE 1279,600
490 PLOT 85,1279,1023
500 MOVE 0,600
510 PLOT 85,0,1023
520 MOVE H%,L%
530 GCOL 0,0
540 VDU 255
550 PROC PB
560 ENDPROC
570 DEF PROC PB
580 MOVE B%,A%
590 GCOL 3,2
600 VDU 254
610 ENDPROC
620 DEF PROC HB
630 I%=I%+1
640 GCOL 3,1
650 MOVE 0,32
660 PRINT"Horizontal?"
670 Z%=VALGET$
680 IF Z%=0 THEN 670
690 Z%=Z%*16
700 SOUND 1,-10,200,2
710 MOVE 0,32
720 PRINT"Horizontal?"

```

```

730 MOVE 0,32
740 PRINT"Vertical?"
750 V%=VALGET#
760 IF V%=0 THEN 750
770 SOUND 1,-10,200,2
780 MOVE 0,32
790 PRINT"Vertical?"
800 ENDPROC
810 DEF PROCMB
820 D%=(B%>H%)-(B%<H%)
830 FOR C%=1 TO V%
840     PROCPB
850     A%=A%+16*V%
860     B%=B%+Z%*D%
870     PROCSE
880     PROCPB
890     PROCGP
900     NEXT
910 FOR C%=1 TO V%
920     PROCPB
930     A%=A%-16*V%
940     B%=B%+Z%*D%
950     PROCSE
960     PROCPB
970     PROCGP
980     NEXT
990 ENDPROC
1000 DEF PROCGP
1010 SOUND 1,-10,A%*B% DIV 2000,5
1020 T=TIME
1030 REPEAT
1040     UNTIL TIME-T>25
1050 ENDPROC
1060 DEF PROCBH
1070 VDU 19,0,2,0,0,0
1080 MOVE 0,500
1090 GCOL 0,3
1100 PRINT"You did it in ";I%
1110 DEF PROCNG

```

```

1120 PRINT''' "Again? (Y/N)"
1130 REPEAT UNTIL INKEY(-69) OR INKEY(-
86)
1140 IF INKEY(-69) THEN RUN
1150 *FX15,0
1160 CLS
1170 ENDPROC
1180 DEF PROCSE
1190 IF A%>0 AND A%<1023 AND B%>0 AND B
%<1280 THEN ENDPROC
1200 SOUND 0,-15,4,20
1210 GCOL 0,3
1220 MOVE 0,500
1230 PRINTE$(RND(3))
1240 PROCNG
1250 GOTO 130
>

```


3D INVADERS

```
0 REM 3D INVADERS
10 MODE 7
20 PROCIN
30 MODE 2
40 VDU 5
50 PROCVA
60 PROCBD
70 PROCIV
80 GCOL 3,1
90 REPEAT
100   FOR P%=0 TO D
110     MOVE U%,64
120     PRINTT$
130     PROCMV
140     MOVE U%,64
150     PRINTS$
160     IF INKEY(-74) THEN PROCLL
170     IF L% THEN PROCFI
180     NEXT
190   PROCAD
200   IF A%=8 THEN A%=7
210   UNTIL I(A%,0)=9
220 PROCEX
230 MODE 6
240 PROCNG
250 RUN
260 END
270 DEF PROCIN
280 VDU 31,10,10,141
290 PRINT "3D Invaders"
300 VDU 31,10,11,141
310 PRINT "3D Invaders"
320 T=TIME
330 REPEAT
340   UNTIL INKEY(-99) OR TIME-T>200
350 ENDPROC
```

```

360 DEF PROCVA
370 DIM C$(7)
380 FOR A%=0 TO 7
390   C$(A%)=CHR$18+CHR$0+CHR$A%
400   NEXT
410 VDU 23,255,60,126,0,0,0,24,24,24
420 VDU 23,254,0,0,255,219,126,36,66,1
29
430 DIM I$(7)
440 FOR A%=2 TO 7
450   I$(A%)=C$(A%-1)+CHR$255+CHR$8+C$
(A%)+CHR$254
460   NEXT
470 I$(0)=C$(7)+CHR$255+CHR$8+C$(1)+CH
R$254
480 I$(1)=C$(3)+CHR$255+CHR$8+C$(1)+CH
R$254
490 VDU 23,253,0,0,0,0,0,102,129,129
500 VDU 23,252,0,0,0,24,24,24,24,0
510 VDU 23,251,0,0,0,0,0,102,126
520 S#=C$(1)+CHR$252+CHR$8+C$(2)+CHR$2
51+CHR$8+C$(3)+CHR$253
530 DIM I(7,2)
540 FOR A%=0 TO 7
550   I(A%,0)=1
560   I(A%,1)=RND(1200)
570   I(A%,2)=RND(512)+300
580   NEXT
590 U%=640
600 M%=32
610 L%=0
620 VDU 23,250,0,0,0,24,24,126,255,255
630 T#=C$(0)+CHR$250
640 D=200
650 VDU 23,249,60,126,255,219,126,60,9
0,153
660 V#=C$(0)+CHR$249
670 K%=0

```

```

680 ENVELOPE 1,138,0,-4,4,1,2,2,100,0,
0,-50,100,100
690 S%=0
700 ENDPROC
710 DEF PROCIV
720 FOR A%=0 TO 7
730   MOVE I(A%,1),I(A%,2)
740   IF I(A%,0)=9 THEN ENDPROC
750   IF I(A%,0) THEN PRINTI$(8-I(A%,0
))
760   NEXT
770 ENDPROC
780 DEF PROCBD
790 MOVE 160,1023
800 FOR A%=7 TO 0 STEP -1
810   PRINTI$(A%);
820   VDU 10,10,8,56-A%,9,11,11
830   NEXT
840 MOVE 288,896
850 GCOL 3,6
860 PRINT"Score: 0"
870 ENDPROC
880 DEF PROCAD
890 SOUND 1,1,250-D,10
900 E%=0
910 D=D-2
920 FOR A%=0 TO 7
930   I(A%,0)=I(A%,0)+1
940   E%=(I(A%,0)=8)+E%
950   NEXT
960 PROCIV
970 ENDPROC
980 DEF PROCFI
990 PROCBU
1000 IF H%>900 THEN L%=0:ENDPROC
1010 IF INKEY(-1) THEN 1190
1020 H%=H%+16
1030 PROCBU
1040 ENDPROC

```

```

1050 DEF PROCCLL
1060 IF L% THEN ENDPROC
1070 SOUND 0,-10,4,-1
1080 L%=-1
1090 B%=U%
1100 H%=128
1110 D=D-RND(1)
1120 PROCBU
1130 ENDPROC
1140 DEF PROCBU
1150 GCOL 3,1
1160 MOVE B%,H%
1170 VDU 252
1180 ENDPROC
1190 L%=0
1200 Q%=0
1210 *FX15,0
1220 FOR A%=0 TO 7
1230   X%=ABS(I(A%,1)-B%)
1240   Y%=ABS(I(A%,2)-H%)
1250   IF X%<32 AND Y%<64 THEN Q%=-1:W%
=A%
1260   NEXT A%
1270 IF Q%=0 THEN ENDPROC
1280 MOVE I(W%,1),I(W%,2)
1290 PRINTV$
1300 GCOL 3,6
1310 MOVE 736,896
1320 PRINT ;K%
1330 K%=K%+(8-I(W%,0))*25
1340 MOVE 736,896
1350 PRINT ;K%
1360 I(W%,1)=RND(1200)
1370 I(W%,2)=RND(512)+300
1380 I(W%,0)=0
1390 ENDPROC
1400 DEF PROCMV
1410 U%=U%+(INKEY(-98)-INKEY(-83))*16
1420 IF U%>1200 THEN U%=0

```

```

1430 IF UZ<0 THEN UZ=1200
1440 ENDPROC
1450 DEF PROC EX
1460 *FX15,0
1470 FOR AZ=1 TO 200
1480   VDU 19,0,RND(7),0,0,0
1490   SOUND RND(3),-12,RND(200),1
1500   NEXT
1510 ENDPROC
1520 DEF PROC NG
1530 PRINT "You let them get within rang
e to fire!!!"
1540 IF HZ<K% THEN HZ=K%
1550 PRINTTAB(0,4); "You scored ";K%
1560 PRINT "Hi-Score ";HZ
1570 PRINT "'Press Y or N"
1580 PRINT "to play again"
1590 REPEAT
1600   UNTIL INKEY(-69) OR INKEY(-86)
1610 IF INKEY(-69) THEN ENDPROC
1620 *FX15,0
1630 CLS
1640 END
1650 ENDPROC

```

>

ANIMATE

```
0 REM ANIMATE
10 ON ERROR GOTO 430
20 MODE 7
30 PRINT TAB(10,10);CHR$(141);"Animat
e"
40 PRINT TAB(10);CHR$(141);"Animate"
50 A=INKEY(100)
60 MODE 2
70 FOR A=1 TO 15
80   VDU 19,A,0,0,0,0
90   NEXT A
100 FOR F=1 TO 15
110   VDU 19,F,7,0,0,0
120   COLOUR F
130   GCOL 0,F
140   PRINT TAB(0,30);"Frame ";F
150   M=2
160   X=640
170   Y=512
180   IF M=2 THEN PLOT 69,X,Y
190   IF M=1 THEN PLOT 69,X,Y:PLOT 71,
X,Y
200   IF INKEY(-35)=-1 THEN M=1
210   IF INKEY(-102)=-1 THEN M=2
220   IF INKEY(-51)=-1 THEN M=3
230   X=X+INKEY(-26)-INKEY(-122)
240   Y=Y+INKEY(-42)-INKEY(-58)
250   IF INKEY(-2)=-1 THEN 280
260   IF INKEY(-72)=-1 THEN 320
270   GOTO 180
280   VDU 19,F,0,0,0,0
290   REPEAT UNTIL INKEY(-2)=0
300   PRINT TAB(0,30);"
310   NEXT F
320 PRINT TAB(0,30);"
330 S=200
```

```

340 FOR N=1 TO F
350   VDU 19,N,7,0,0,0
360   FOR A=1 TO S:NEXT A
370   IF INKEY(-74)=-1 THEN 370
380   IF INKEY(-68)=-1 THEN S=S-10
390   IF INKEY(-82)=-1 THEN S=S+10
400   VDU 19,N,0,0,0,0
410   NEXT N
420 GOTO 340
430 MODE 4
440 REPORT:PRINT" at line ";ERL
450 END

```

>

ASPIRIN

```
0 REM ASPIRIN
10 MODE 6:VDU 19,0,4,0,0,0
20 MODE 7
30 PROCIN
40 PROCVA
50 PROCUS
60 MODE 5
70 VDU 19,2,10,0,0,0
80 VDU 5
90 GCOL 3,1
100 PROCMZ
110 PROCPP
120 REPEAT
130   PROCPP
140   U%=U%+(INKEY(-98)-INKEY(-83))*8
150   P%=P%+(INKEY(-105)-INKEY(-73))*4
160   I%=POINT(U%,P%)=1
170   PROCPP
180   UNTIL I%
190 T=TIME
200 REPEAT
210   PROCPP
220   REPEAT
230     SOUND 1,-6,P%/5,2
240     PROCPP
250     PROCPP
260     V%=(INKEY(-98)-INKEY(-83))*8
270     Q%=(INKEY(-105)-INKEY(-73))*4
280     PROCPP
290     PROCPP
300     UNTIL POINT(U%+V%,P%+Q%)=1
310     U%=U%+V%
320     P%=P%+Q%
330     PROCPP
340     UNTIL P%=8
350 T=TIME-T
```



```

360 IF H%>T THEN H%=T:C%=S%
370 VDU 19,0,4,0,0,0
380 VDU 4
390 VDU 31,2,5
400 PRINT ;T/100;" seconds"
410 VDU 31,2,20
420 PRINT "stomach ";S%
430 VDU 31,2,28
440 PRINT "best so far "
450 PRINT TAB(2);H%/100;" seconds"
460 PRINTTAB(1);"(stomach ";C%;")"
470 REPEAT
480   UNTIL INKEY(-99)
490 T=TIME-T
500 *FX15,0
510 RUN
520 END
530 DEF PROCIN
540   VDU 31,10,10,141
550   PRINT"Aspirin"
560   VDU 31,10,11,141
570   PRINT"Aspirin"
580   T=TIME
590   REPEAT
600     UNTIL INKEY(-99) OR TIME-T>200
610   CLS
620   INPUT "Stomach Number ",S%
630   T=RND(-S%)
640   ENDPROC
650 DEF PROCVA
660   DIM A$(3)
670   A$(0)=CHR$255+CHR$11
680   A$(1)=CHR$255
690   A$(2)=CHR$254+CHR$10
700   A$(3)=CHR$254
710   B$=""
720   N%=0
730   FOR A%=1 TO 84
740     IF N%=0 THEN N%=RND(2)-1:GOTO 78

```

```

750     IF N%=1 THEN N%=2+RND(2)-1:GOTO
780
760     IF N%=2 THEN N%=2+RND(2)-1:GOTO
780
770     IF N%=3 THEN N%=RND(2)-1:GOTO 78
0
780     B$=B$+A$(N%)
790     NEXT
800 L%=1
810 U%=640
820 P%=1020
830 I%=-1
840 ENDPROC
850 DEF PROCUS
860 VDU 23,255,1,2,4,8,16,32,64,128
870 VDU 23,254,128,64,32,16,8,4,2,1
880 ENDPROC
890 DEF PROCMZ
900 L%=1
910 M%=1
920 REPEAT
930     M%=M%+1
940     PROCPL
950     L%=L%+1
960     IF L%=64 THEN L%=1
970     PROCPL
980     UNTIL M%=40
990 ENDPROC
1000 DEF PROCPL
1010 PRINT MID$(B$,L%,30)
1020 ENDPROC
1030 DEF PROCPP
1040 GCOL 3,2
1050 MOVE U%,P%+4
1060 DRAW U%,P%-4
1070 PLOT 69,U%-8,P%
1080 PLOT 69,U%+8,P%
1090 ENDPROC

```

>

BALLOONS

```
0 REM BALLOONS
10 MODE 7
20 PROCIN
30 MODE 2
40 VDU 5
50 PROCVA
60 GCOL 3,5
70 *FX10,5
80 *FX9,5
90 SOUND 3,3,3,-1
100 PROCUS
110 PROCSH
120 REPEAT
130   PROCSH
140   VDU 4
150   VDU 30,11
160   VDU 5
170   PROCPC
180   IF POINT(U%+32,P%-32)<>0 THEN FR
OCEN
190   U%=U%+(INKEY(-98)-INKEY(-83))*32
200   PROCSH
210   MOVE RND(1280),991
220   GCOL 3,RND(7)
230   IF RND(100)>90 THEN GCOL 3,9
240   PRINT STRING$(RND(S% DIV 100)+1,
CHR$(255))
250   PROCPC
260   S%=S%+1
270   UNTIL FALSE
280 DEF PROCVA
290 U%=640
300 ENVELOPE 1,1,1,-1,1,25,50,25,127,0
,0,-5,126,126
310 ENVELOPE 2,1,100,100,200,100,50,50
,127,0,0,-5,126,126
```

```

320 ENVELOPE 3,25,0,3,0,2,1,0,50,0,-50
,0,50,50
330 P%=900
340 S%=0
350 ENDPROC
360 DEF PROCSH
370 MOVE U%,P%
380 PRINTS$
390 ENDPROC
400 DEF PROCEN
410 P%=P%-32
420 IF POINT(U%+32,P%)=9 THEN P%=P%+32
*RND(4):SOUND 0,1,100,10:ENDPROC
430 IF P%>991 THEN P%=991
440 SOUND RND(2),-15,P% DIV 10,10
450 IF P%>32 THEN ENDPROC
460 VDU 4
470 VDU 19,0,6,0,0,0
480 COLOUR 5
490 IF H%<S% THEN H%=S%
500 PRINT TAB(5,3);"Hi-Score ";H%
510 COLOUR 4
520 PRINT TAB(5,15);"Score ";S%
530 COLOUR 9
540 PRINT TAB(1,25);"Hit any key kiddo
"
550 FOR A=1 TO 2000
560 NEXT
570 *FX15,0
580 A=GET
590 RUN
600 DEF PROCIN
610 VDU 31,10,10,141
620 PRINT"Balloons"
630 VDU 31,10,11,141
640 PRINT"Balloons"
650 T=TIME
660 REPEAT
670 UNTIL INKEY(-99) OR TIME-T>200

```

```

680 ENDPROC
690 DEF PROCUS
700 VDU 23,255,126,255,255,255,255,255
,255,126
710 VDU 23,254,24,24,60,0,0,0,0,0
720 VDU 23,253,0,0,0,126,68,0,0,0
730 VDU 23,252,0,0,0,0,60,102,195,195
740 FOR A%=1 TO 14
750     READ B%
760     S%=S%+CHR$(B%)
770     NEXT
780 ENDPROC
790 DATA 18,3,7,254,8,18,3,1,253,8,18,
3,13,252
800 DEF PROCPC
810 GCOL 3,7
820 MOVE 0,900
830 PRINT ;S%
840 ENDPROC

```

>

BATTLEFIELD

```

      0 REM BATTLEFIELD
      10 MODE7
      20 PRINTTAB(10,10);CHR$(141);"Battlef
ield"
      30 PRINTTAB(10);CHR$(141);"Battlefiel
d"
      40 A=INKEY(100)
      50 MODE1
      60 I=0
      70 J=0
      80 VDU 23,240,255,255,255,255,255,255
,255,255
      90 VDU 19,2,5,0,0,0,19,3,6,0,0,0
     100 DATA "*123456789*"
     110 DATA "1X*XX*XX*XX1"
     120 DATA "2*XX*XX*XX*2"
     130 DATA "3X*XX*XX*XX3"
     140 DATA "4* * * * *4"
     150 DATA "5 * * * * 5"
     160 DATA "6*0*0*0*0*6"
     170 DATA "70*0*0*0*07"
     180 DATA "8*0*0*0*0*8"
     190 DATA "*123456789*"
     200 DIM S$(10,11)
     210 FOR A=1 TO 10
     220   READ B$
     230   FOR B=1 TO 11
     240     IF MID$(B$,B,1)="*" THEN S$(A,
B)=CHR$(240) ELSE S$(A,B)=MID$(B$,B,1)
     250   NEXT B,A
     260   COLOUR 1
     270   PRINT"Your score ";I;"Computer's
score ";J
     280   FOR A=1 TO 10
     290     FOR B=1 TO 11
```

```

300         IF S$(A,B)="X" THEN COLOUR 1
:SOUND 1,-15,0,1
310         IF S$(A,B)="O" THEN COLOUR 2
:SOUND 1,-15,200,1
320         IF S$(A,B)<>"O" AND S$(A,B)<
>"X" THEN COLOUR 3:SOUND 1,-15,50,1
330         PRINT TAB(B+15,A+5);S$(A,B)
340         NEXTB
350         NEXTA
360         PRINTTAB(0,18);
370         PRINT"Your move"
380         INPUT"From "A,B
390         INPUT"To "C,D
400         PRINTTAB(0,18); " "
410         PRINT " "
420         PRINT " "
430         PRINT " "
440         IF ABS(A-C)=1 AND ABS(B-D)=1 THE
N 470
450         PRINT"Do not try to cheat!"
460         GOTO 360
470         IF S$(C+1,D+1)="X" THEN I=I+1:FO
R S=0 TO 200 STEP 5:SOUND 1,-15,S,1:NEXT
S
480         S$(A+1,B+1)=" "
490         S$(C+1,D+1)="O"
500         FL=FNMOVE("O")
510         IF FL=1 THEN J=J+1:FOR S=200 TO
O STEP -5:SOUND 1,-15,S,1:NEXTS:GOTO 530
520         FL=FNMOVE(" ")
530         S$(E,F)=" "
540         S$(E+G,F+H)="X"
550         VDU30
560         IF I=13 OR J=13 THEN 730
570         GOTO 260
580         DEF FNMOVE(A$)
590         E=2
600         F=2
610         G=0

```

```

620     H=0
630     IF S$(E,F)<>"X" THEN 690
640     IF S$(E+1,F+1)=A$ THEN G=1:H=1
650     IF S$(E+1,F-1)=A$ THEN G=1:H=-1
660     IF S$(E-1,F+1)=A$ THEN G=-1:H=1
670     IF S$(E-1,F-1)=A$ THEN G=-1:H=-1
680     IF G<>0 AND H<>0 THEN =1
690     E=E+1
700     IF E>10 THEN E=2:F=F+1
710     IF F>11 THEN =0
720     GOTO 630
730     CLS
740     PRINTTAB(0,10);
750     IF I=13 AND J<>13 THEN PRINT"We l
1 done-you won!"
760     IF J=13 AND I<>13 THEN PRINT"Ha
Ha! I thought that I would beat you!"
770     IF I=13 AND J=13 THEN PRINT"What
a game-it was a draw"
780     PRINT
790     INPUT"Another go",G$
800     IF LEFT$(G$,1)="Y" THEN RUN
810     END

```

>

BIG BUSINESS

```
0 REM BIG BUSINESS
10 ON ERROR RUN
20 MODE7
30 PRINTTAB(10,10);CHR$(141);"Big Bus
iness"
40 PRINTTAB(10);CHR$(141);"Big Busine
ss"
50 A=INKEY(100)
60 PROC_KEYS
70 CLS
80 MODE7
90 DIMC$(10),V(30,10),H(10)
100 M=1000
110 @%=&2020A
120 FOR C=1 TO 10
130   V(0,C)=RND(320)/100
140   V(1,C)=RND(320)/100
150   READ C$(C)
160   NEXT C
170 DATA Corna,Ratai,Dargon,Harps,Atnd
y,Texsa,Inclairs,Commdora,Alep,Comnas
180 ON ERROR GOTO 200
190 FOR Y=1 TO 29
200   REM*****TABLE*****
210   CLS
220   PRINT"Year ";
230   Y$=STR$(Y)
240   PRINTY$
250   PRINT"-----
-----"
260   PRINT"Company","Price","No. ","To
tal "
270   PRINT"-----
-----"
```

```

280   FORC=1 TO 10:PRINT"f";STR$(C-1);
":";C$(C);TAB(12);V(Y,C);TAB(20);H(C);TA
B(30);V(Y,C)*H(C)
290   NEXTC
300   PRINT"-----
-----"
310   PRINT"Money left over ";M
320   AS=0
330   FORC=1 TO 10
340     AS=AS+(H(C)*V(Y,C))
350   NEXTC
360   PRINT"Money in shares ";AS
370   PRINT:PRINT:PRINT
380   PRINT"1.....Buy Shares"
390   PRINT"2.....Sell Shares"
400   PRINT"3.....Move to next
year"
410   IF Y>1 THEN PRINT"4.....
Previous Record"
420   *FX 15,1
430   K=VAL(GET$)
440   IF K<1 OR K>4 THEN 420
450   IF K=1 THEN PROC_BUY
460   IF K=2 THEN PROC_SELL
470   IF K=4 AND Y>1 THEN MODE5:PROC_R
ECORD:MODE7
480   IF K<>3 THEN 210
490   GOTO 1200
500   DEF PROC_BUY
510   CLS
520   PRINTTAB(10);"Share Purchase For
m"
530   PRINT"-----
-----"
540   PRINT"Company"
550   PRINT"-----
-----"
560   PRINT"Each share will cost "

```

```

570 PRINT"-----
-----"
580 PRINT"Quantity required"
590 PRINT"-----
-----"
600 PRINT"TOTAL"
610 PRINT"=====
=====
620 INPUTTAB(7,2),M$
630 N=0
640 FOR V=1 TO 10
650     IFC$(V)=M$ THEN N=V
660     NEXT V
670 IF N=0 THEN PRINTTAB(7,2); "
        ":GOTO 620
680 PRINTTAB(20,4); " ` "; V(Y,N)
690 INPUTTAB(17,6),Q
700 IF Q*V(Y,N)>M THEN PRINT TAB(17,
6); "        ":GOTO 690
710 PRINTTAB(6,8); " ` "; Q*V(Y,N)
720 M=M-(Q*V(Y,N))
730 H(N)=H(N)+Q
740 FOR A=1 TO 4000:NEXT
750 ENDPROC
760 DEF PROC_SELL
770 CLS
780 PRINTTAB(10); "Share sales form"
790 PRINT"-----
-----"
800 PRINT"Company"
810 PRINT"-----
-----"
820 PRINT"Each share will sell for `
"
830 PRINT"-----
-----"
840 PRINT"Quantity to be sold"
850 PRINT"-----
-----"

```

```

860 PRINT "TOTAL  \ "
870 PRINT "=====
===== "
880 INPUT TAB(7,2),M$
890 N=0
900 FOR V=1 TO 10
910     IFC$(V)=M$ THEN N=V
920     NEXT V
930 IF N=0 THEN PRINT TAB(7,2); "
           ":GOTO 620
940 PRINT TAB(26,4);V(Y,N)
950 INPUT TAB(19,6),Q
960 IF Q>H(N) THEN PRINT TAB(19,6); "
           ":GOTO 950
970 T=Q*V(Y,N)
980 H(N)=H(N)-Q
990 PRINT TAB(7,8);T
1000 M=M+T
1010 FORA=0 TO 4000:NEXTA
1020 ENDPROC
1030 DEF PROC_RECORD
1040 CLS
1050 PRINT "-----"
1060 PRINT "Company"
1070 PRINT "-----"
1080 INPUT TAB(7,1),M$
1090 N=0
1100 FOR V=1 TO 10
1110     IFC$(V)=M$ THEN N=V
1120     NEXTV
1130 IF N=0 THEN PRINT TAB(7,1); "
           ":GOTO1080
1140 MOVE 0,V(1,N)*100
1150 FOR Y1=2 TO Y
1160     DRAW Y1*30,V(Y1,N)*100
1170     NEXTY1
1180 REPEAT UNTIL GET$(">")
1190 ENDPROC
1200 FOR C=1 TO 10

```

```

1210      IF V(Y,C)=0 THEN I=0:GOTO 1320
1220      A=V(1,C)
1230      FOR Y1=1 TO Y
1240          A=A+(V(Y1,C)-V(Y1-1,C))
1250      NEXT Y1
1260      A=A/(Y1-1)
1270      N=A*RND(100)+50
1280      D=ABS(N)
1290      I=RND(D*2.5)-D
1300      I=I/100
1310      IF N<0 AND RND(3)<>1 THEN I=-I

1320      V(Y+1,C)=V(Y,C)+I
1330      NEXT C
1340      FOR C=1 TO 10
1350          IF V(Y+1,C)<0 THEN V(Y+1,C)=0
1360      NEXT C,Y
1370      CLS
1380      PRINT"That is your thirty years
up!"
1390      PRINT"Didn't it pass quickly!"
1400      PRINT"Your profits over the 30 y
ear period"
1410      PRINT"were  "AS+M-1000
1420      P=AS+M-1000
1430      IF P<0 THEN PRINT"How did you ma
nage to make a loss?"
1440      IF P<1000 AND P>0 THEN PRINT"At
least you made a profit"
1450      IF P<2000 AND P>1000 THEN PRINT"
That is not bad"
1460      IF P>3000 THEN PRINT"Very good-p
ity you didn't really""make  ";P
1470      INPUT"Another go",G$
1480      IF LEFT$(G$,1)="Y" THEN RUN
1490      END
1500      DEF PROC_KEYS
1510      *KEY 0 CornalM
1520      *KEY 1 RataiM

```

1530 *KEY 2 DargonIM
1540 *KEY 3 HarpsIM
1550 *KEY 4 AtndyIM
1560 *KEY 5 TexsaIM
1570 *KEY 6 InclairsIM
1580 *KEY 7 CommdoraIM
1590 *KEY 8 AplepIM
1600 *KEY 9 ComnasIM
1610 ENDPROC

>

BREAKOUT

```
0 REM BREAKOUT
10 MODE7
20 PRINTTAB(10,10);CHR$(141);"Breakout"
t"
30 PRINTTAB(10);CHR$(141);"Breakout"
40 A=INKEY(100)
50 MODE 2
60 DIM P(7,18)
70 FOR A=1 TO 7
80   FOR B=1 TO 18
90     P(A,B)=1
100    NEXT B,A
110   VDU 23;8202;0;0;0;
120   VDU 23,244,0,254,254,254,254,254
,254,254
130   VDU 23,245,255,255,255,255,0,0,0
,0
140   VDU 23,246,255,255,255,255,255,2
55,255,255
150   COLOUR 2
160   FOR A=1 TO 20
170     PRINTCHR$(246);
180     NEXT
190   FOR A=1 TO 19
200     PRINTCHR$(246);SPC(18);CHR$(24
6);
210     NEXT A
220   FOR A=1 TO 7
230     COLOUR A
240     FOR B=1 TO 18
250       PRINTTAB(B,A+2);CHR$(244);
260       NEXT B
270     NEXT A
280   Y=19:X=RND(18)
290   B=9
300   XI=1
```

```

310    YI=-1
320    Q=0
330    C=1
340    PRINT TAB(B,19);CHR$(245);CHR$(2
45);CHR$(245)
350    PRINT TAB(X,Y);"O"
360    R=X
370    S=Y
380    N=B
390    X=X+XI
400    Y=Y+YI
410    IF Y<0 THEN YI=-YI
420    IF Y=19 AND ( B-1=X OR B=X OR B+
1=X OR B+2=X OR B+3=X) THEN YI=-YI:Y=18
430    IF Y>=20 THEN PROC_LOST
440    IF X<2 OR X>17 THEN XI=-XI:GOTO
390
450    IF Y=1 THEN YI=-YI:GOTO 390
460    IF Y>2 AND Y<9 THEN PROC_CHECK
470    PRINT TAB(0,25);"Score ";Q
480    PRINT "Ball ";C
490    B=B+INKEY(-98)-INKEY(-83)
500    IF B<1 THEN B=1
510    IF B>16 THEN B=16
520    PRINT TAB(R,S);" "
530    PRINT TAB(N,19);" "
540    GOTO 340
550    DEF PROC_CHECK
560    A=Y-2
570    IF P(A,X)=0 THEN ENDPROC
580    P=9-Y
590    P(A,X)=0
600    SOUND 1,-15,50*P,5
610    Q=Q+P*10
620    YI=-YI
630    ENDPROC
640    DEF PROC_LOST
650    SOUND 0,-15,2,5
660    C=C+1

```



```

670   IF C<11 THEN Y=19:X=RND(18):YI=-
1:FOR D=1 TO 1000:NEXT D:ENDPROC
680   CLG
690   PRINT "Score ";Q
700   *FX 15,1
710   PRINT "Another go":A#=GET$
720   IFLEFT$(A$,1)="Y" THEN RUN
730   END

```

>

THE CLEVER COMPUTER

```

      0 REM THE CLEVER COMPUTER
    10 MODE 7
    20 PRINTCHR$(141);"The Clever Compute
r"
    30 PRINTCHR$(141);"The Clever Compute
r"
    40 PRINT
    50 DIM K$(30),R$(94):PROC_DATA
    60 PRINT"Hello, how are you today"
    70 INPUT LINE"",H$:IF H$="GOODBYE" TH
EN END
    80 H$=" " +H$+" "
    90 L=0:REPEAT L=L+1:UNTIL INSTR(H$,K$
(L))<>0 OR L=30
   100 IFL=30THEN L=31
   110 R=L*3-RND(3)+1
   120 PRINT R$(R):PRINT:PRINT
   130 GOTO70
   140 DEF PROC_DATA
   150 FOR A=1 TO 30:READ K$(A):K$(A)=" "
+K$(A)+" ":NEXTA
   160 FOR A=1 TO 94:READ R$(A):NEXT A
   170 ENDPROC
   180 REM DATA FOR PROGRAM.
   190 REM KEYWORD DATA
   200 DATA HELLO,GOOD,BAD,OK,HOW,WHAT,IF
,I,YOU,AWFUL,OUCH,COMPUTER,BBC,UNTIL,TRU
E,WRONG,RIGHT,PROBABLY,SOUND,COLOUR,GRAP
HIC,CASSETTE,FRIEND,NAME,CLEVER,STUPID,T
HICK,PRINTER,GO,JUMP
   210 DATA You are a very friendly perso
n.
   220 DATA You will always be my friend.
   230 DATA I like you!
   240 DATA What a positive attitude you
take
```

250 DATA It may not be as good as you
think
260 DATA I am glad to hear that!
270 DATA That is a shame.
280 DATA I am very sorry to hear that.
290 DATA Why is that?
300 DATA Only OK?
310 DATA Oh it can't be that bad can i
t?
320 DATA Well that's not too bad.
330 DATA Don't be so worried about tha
t!
340 DATA That is none of your business
350 DATA That is TOP SECRET informatio
n!
360 DATA What is it to you anyway?
370 DATA Must I answer that?
380 DATA Don't get too concerned about
me friend!
390 DATA My IF statement resides at &F
E09 in my ROM!
400 DATA What would happen if that wer
e not so?
410 DATA So what?
420 DATA Your approach to life is very
selfish
430 DATA I am not interested in your s
illy little problems!
440 DATA I would forget that idea!
450 DATA Do I worry you?
460 DATA Are you trying to find out ab
out me?
470 DATA I am only a computer. I am no
t human.
480 DATA What is wrong?
490 DATA That is sad!
500 DATA My CPU is sore too!

510 DATA Is it very sore?
520 DATA My dynamic RAM is in pain too
!
530 DATA Is there anything I could do
to help?
540 DATA Do you like computers that mu
ch?
550 DATA I am a computer
560 DATA Computers help you.
570 DATA BBC Computers are the best!
580 DATA Why did you buy a me anyway?
590 DATA Could you tell me what's on T
V?
600 DATA Until when?
610 DATA Do you mean REPEAT UNTIL FALS
E
620 DATA I understand
630 DATA Surely not?
640 DATA Never!
650 DATA What cheek!
660 DATA I am never wrong!
670 DATA No that is quite right
680 DATA You could be locked up for th
at!
690 DATA I agree.
700 DATA That's fine by me!
710 DATA I am glad to hear that sir!
720 DATA Be more certain!
730 DATA Yes it could couldn't it?
740 DATA The chances of that are 100-1
!
750 DATA I have never heard my sound!
760 DATA It is loud!
770 DATA Yes I hear it faintly.
780 DATA I have 16 colours.
790 DATA Are you talking about a rainb
ow?
800 DATA I like lots of colours!

810 DATA Have you seen my graphics?
820 DATA MODE 0 graphics are very complicated.
830 DATA Did you buy me because of my graphics.
840 DATA Discs are better than cassettes.
850 DATA Do cassettes give you problems.
860 DATA I will see what I can do about it.
870 DATA I am your friend!
880 DATA I am glad that you are my friend.
890 DATA Yes, let's be friends!
900 DATA By the way what is your name?
910 DATA My name is the BBC Computer.
920 DATA I can never remember names!
930 DATA I am clever!
940 DATA How intelligent!
950 DATA That depends on what you mean by clever.
960 DATA At least I am not stupid.
970 DATA Humans are very stupid.
980 DATA Very stupid?
990 DATA Thick? How thick?
1000 DATA That was thick!
1010 DATA That depends on what you mean by thick.
1020 DATA My printer is sick.
1030 DATA Do I even have a printer?
1040 DATA Is my printer RS423 or Cetronics?
1050 DATA Go where?
1060 DATA I would be too scared to go there!
1070 DATA You must not go. It is dangerous!
1080 DATA Jumps are dangerous.

1090 DATA I could jump 100 busses!
1100 DATA Off a cliff?
1110 DATA What does that mean?
1120 DATA I do not understand.
1130 DATA I am confused!
1140 DATA That is very interesting.
>

'COPTER

```
      0 REM COPTER
      10 VDU 23,255,255,24,24,60,129,255,24
,231
      20 MODE 7
      30 VDU 31,10,10,141
      40 PRINT"Copter"
      50 VDU 31,10,11,141
      60 PRINT"Copter"
      70 VDU 31,4,20
      80 PRINT"Speed Difficulty? 115 (5 eas
iest)"
      90 REPEAT
      100   D=GET
      110   UNTIL D>48 AND D<54
      120 MODE 2
      130 SOUND 0,-15,3,-1
      140 PROCDB
      150 S%=0
      160 VDU 5
      170 *FX9,1
      180 *FX10,1
      190 D=(D-49)*20
      200 ENVELOPE 1,1,0,0,0,60,60,60,100,2,
0,-1,100,126
      210 R%=10
      220 REPEAT
      230   GCOL 0,0
      240   U%=680
      250   F%=1024
      260   REPEAT
      270     PROCCP
      280     GCOL 0,2
      290     P%=F%-16
      300     U%=U%+(INKEY(-98)-INKEY(-83))*
16
      310     P%=P%-(INKEY(-73))*32
```

```

320     IF U%<0 THEN U%=0
330     IF U%>1248 THEN U%=1248
340     IF P%>1024 THEN P%=1024
350     PROCCP
360     S%=S%+INKEY(-73)*35
370     PROCCP
380     GCOL 0,0
390     UNTIL P%=544
400     IF ABS((A(R%)-32)-U%)>24 THEN PR
OCEX:MODE 4:PROCEN
410     PROCCP
420     S%=S%+A(R%)
430     MOVE A(R%),512
440     GCOL 0,3
450     DRAW A(R%),R%*32
460     GCOL 0,2
470     R%=R%-1
480     UNTIL R%=0
490     MODE 7
500     *FX15,0
510     VDU 31,16,12,141,132
520     PRINT "WELL DONE"
530     VDU 31,16,13,141,133
540     PRINT "WELL DONE"
550     IF H%<S% THEN H%=S%
560     VDU 31,12,17
570     PRINT"You scored.....";S%
580     VDU 31,13,22
590     PRINT"Speed Difficulty ";D/20+1
600     VDU 31,14,20
610     PRINT"Hi-Score ";H%
620     K=GET:RUN
630     PROCEX
640     DEF PROCDB
650     GCOL 0,3
660     MOVE 1280,0
670     PLOT 85,1280,512
680     MOVE 0,0
690     PLOT 85,0,512

```



```

700 GCOL 0,9
710 DIM A(10)
720 FOR A=1 TO 10
730   REPEAT
740     P=RND(1280)
750     UNTIL POINT(P,512)=3
760     MOVE P,512
770     DRAW P,A*32
780     A(A)=P
790   NEXT
800 ENDPROC
810 DEF PROCCP
820 VDU 25,4,U%;P%;255
830 ENDPROC
840 DEF PROCGP
850 FOR W=1 TO D
860   NEXT
870 ENDPROC
880 DEF PROCSC
890 MOVE 450,50
900 PRINT S%
910 ENDPROC
920 DEF PROCEX
930 SOUND 16,1,6,20
940 VDU 19,3,8,0,0,0
950 VDU 19,0,8,0,0,0
960 T=TIME
970 REPEAT
980   UNTIL TIME-T>20
990 VDU 20
1000 FOR A%=0 TO 128 STEP 2
1010   GCOL 0,RND(8)
1020   DRAW FNR(U%),FNS(P%)
1030   NEXT
1040 MOVE 0,820
1050 PRINT"Your copter has been"
1060 FOR A%=1 TO 20
1070   GCOL 0,RND(15)
1080   MOVE FNR(256),FNS(700)

```

```

1090 PRINT "OBLITERATED"
1100 NEXT
1110 REPEAT
1120 UNTIL INKEY(-99)
1130 ENDPROC
1140 DEF PROCEN
1150 PRINT "Too bad sucker..."
1160 PRINT "You scored ";S%
1170 IF H%<S% THEN H%=S%
1180 IF H%>S% THEN PRINT "You didn't bea
t the ";
1190 PRINT "Hi-Score ";H%
1200 PRINT "' "'Play again? (Y/N)"
1210 REPEAT
1220 UNTIL INKEY(-69) OR INKEY(-86)
1230 IF NOT INKEY(-86) THEN RUN
1240 END
1250 DEF FNR(X%)=X%+32+RND(A%)-RND(A%)
1260 DEF FNS(X%)=X%-16+RND(A%)-RND(A%)

```

>

DEFENDER

```
10 ON ERROR GOTO 1430
20 REM*****
30 REM*      DEFENDER  0.3      *
40 REM*****
50 PROCDEFINE:PROCSETUP
60 MODE7
70 PROCINSTR
80 MODE2
90 VDU23;8202;0;0;0;0;
100 REPEAT
110   PROCMOVE:PROCSCREEN
120   PROCALIEN:IF GRD<4 AND FLG THEN
SOUND 1,3,200,10:FLG=0
130   PROCHECK
140   UNTIL GRD=0
150 MODE7:PROCGAMEEND
160 GOTO60
170 DEFPROCDEFINE
180 VDU23,240,0,0,0,0,0,255,255,0
190 VDU23,241,0,0,0,1,63,25,255,0
200 VDU23,242,0,0,0,128,252,25,255,0
210 VDU23,243,16,56,84,84,254,56,56,16
220 VDU23,245,0,0,0,0,225,30,0,0
230 VDU23,255,255,255,255,255,255,255,
255,255
240 VDU23,246,23,45,243,200,100,73,4,1
250 HIT$=CHR$(246)
260 ENVELOPE1,129,127,1,1,55,1,1,127,-
1,0,-4,126,126
270 ENVELOPE2,1,1,0,0,72,1,1,1,-1,0,-3
,85,85
280 ENVELOPE3,1,129,20,-20,0,5,5,10,12
7,-1,0,-8,110
290 INV$=CHR$(243)
300 LFT$=CHR$(240)+CHR$(241)
310 RGHT$=CHR$(242)+CHR$(240)
```

```

320 LSR$=CHR$(245)
330 INV2$=CHR$(243)+CHR$(10)+CHR$(8)+CHR$(255)+CHR$(10)+CHR$(8)+CHR$(32)
340 ENDPROC
350 DEFPROCSETUP
360 DIMI$(10),IX%(10),IY%(10),IM%(10)
370 FOR T=1 TO 10: I$(T)=INV$: IY%(T)=0: IM%(T)=RND(2): NEXT T: SP$=""
380 C=-1
390 FOR U=1 TO 10: C=C+2: IX%(U)=C: NEXT U
400 IX%(6)=3
410 GRD$=CHR$(255)
420 SCORE=0: HSCORE=0: VERT=8: SHIP$=LFT$: VDU23;8202;0;0;0;
430 ENDPROC
440 DEFPROCSCREEN
450 COLOUR7
460 PRINTTAB(0,1);"
"
470 FOR U=1 TO GRD: PRINTTAB(U,29);GRD$:NEXT U
480 COLOUR6: PRINTTAB(0,0);"SCORE=";SCORE;" FUEL=";GRD
490 COLOUR3
500 PRINTTAB(10,VERT);SHIP$:PRINTTAB(10,VERT-1);" ";TAB(10,VERT+1);" "
510 ENDPROC
520 DEFPROCMOVE
530 A$=INKEY$(0): IFA$="" ENDPROC
540 IF A$="A" OR A$="a" AND VERT>0 THEN VERT=VERT-1
550 IF A$="Z" OR A$="z" AND VERT<28 THEN VERT=VERT+1
560 IF A$="" AND SHIP$=LFT$ THEN SHIP$=RGHT$:GOTO 590
570 IF A$="" AND SHIP$=RGHT$ THEN SHIP$=LFT$
580 IF A$="/" OR A$="?" THEN PROCLASER(1):PROCLASER(0)

```

```

590 *FX 15 0
600 ENDPROC
610 DEFPROCLASER(COL):COLOUR COL
620 IF COL=1 THEN SOUND1,1,100,2
630 IF SHIP$=LFT$ THEN 690
640 FOR LP=12 TO 19
650   FORSCAN%=1TO10: IF VERT=IY%(SCAN%
) AND LP=IX%(SCAN%) AND I$(SCAN%)<>" " T
HEN PROCHIT
660     NEXT SCAN%
670     PRINTTAB(LP,VERT);LSR$;
680     NEXT LP:GOTO 740
690 FOR LP=9 TO 0 STEP-1
700   FORSCAN%=1TO10: IF VERT=IY%(SCAN%
) AND LP=IX%(SCAN%) AND I$(SCAN%)<>" " T
HEN PROCHIT
710     NEXT SCAN%
720     PRINTTAB(LP,VERT);LSR$;
730     NEXT LP
740 ENDPROC
750 DEFPROCALIEN:COLOUR2
760 FORT%=1TO SHEET
770   IFRND(2)>1 AND I$(T%)<>" " THEN I
Y%(T%)=IY%(T%)+IM%(T%)
780   IF IY%(T%)>=28 AND IM%(T%)>0 THE
N IM%(T%)=-1:SOUND1,2,50,10:GRD=GRD-1:I$
(T%)=INV2$
790   IF IY%(T%)=0 THEN I$(T%)=INV$:IM
%(T%)=RND(2):PRINTTAB(IX%(T%),IY%(T%));"
";TAB(IX%(T%),IY%(T%)-1);" ";TAB(IX%(T
%),IY%(T%)-2);" "
800   PRINTTAB(IX%(T%),IY%(T%));I$(T%)
;
810   IF I$(T%)<>INV2$ THEN PRINTTAB(I
X%(T%),IY%(T%)-IM%(T%));" "
820   NEXT
830 ENDPROC
840 DEFPROCHIT: IF COL=0 THEN NUMB=NUMB

```

-1

```

850 PRINTTAB(IX%(SCAN%),IY%(SCAN%));"
"
860 IF COL=1 THEN PROCEXPLOSION:SCORE=
SCORE+50:IF I$(SCAN%)=INV2$THEN SCORE=SC
ORE+50
870 IF I$(SCAN%)=INV2$ THEN PROCDROP
880 IF COL=0 THEN I$(SCAN%)=SP$
890 ENDPROC
900 DEFPROCDROP:IF COL=0 ENDPROC
910 COLOUR7
920 FOR HGT%=IY%(SCAN%) TO 28
930     SOUND1,-15,HGT%*2,1
940     PRINTTAB(IX%(SCAN%),HGT%);CHR$(2
55)
950     A$=INKEY$(10)
960     PRINTTAB(IX%(SCAN%),HGT%);CHR$(3
2)
970     NEXT
980 PRINTTAB(IX%(SCAN%),IY%(SCAN%));"
"
990 PRINTTAB(IX%(SCAN%),HGT%);" "
1000 GRD=GRD+1
1010 COLOUR1:ENDPROC
1020 DEFPROCCHCK:IF NUMB >0 THEN ENDPR
OC
1030 CLS:SHEET=SHEET+1:IF SHEET>10 THEN
SHEET=10
1040 NUMB=SHEET
1050 FORCK=1TO NUMB:I$(CK)=INV$:IY%(CK)
=1:NEXT
1060 ENDPROC
1070 DEFPROCEXPLOSION:COLOUR1
1080 PRINTTAB(IX%(SCAN%),IY%(SCAN%));HI
T$
1090 FOR SND=-15 TO 0 STEP0.5
1100     SOUND0,SND,100,1
1110     NEXT
1120 PRINTTAB(IX%(SCAN%),IY%(SCAN%));"
"

```

```

1130 ENDPROC
1140 DEFPROCINSTR
1150 CLS
1160 PRINTTAB(14,1);CHR$(141);CHR$(131)
;"DEFENDER"
1170 PRINTTAB(14,2);CHR$(141);CHR$(131)
;"DEFENDER"
1180 PRINTTAB(0,4);CHR$(134);"YOU MUST
DEFEND YOUR PLANETS PRECIOUS"
1190 PRINTTAB(0,5);CHR$(134);"POWER CEL
LS FROM THE INVADING ALIENS"
1200 PRINTTAB(0,6);CHR$(134);"YOU ARE I
N COMMAND OF THE YELLOW SPACE-"
1210 PRINTTAB(0,7);CHR$(134);"SHIP.IT I
S ARMED WITH A SINGLE PHASER,"
1220 PRINTTAB(0,8);CHR$(134);"YOU'RE ON
LY WEAPON, WHICH IS OPERATED "
1230 PRINTTAB(0,9);CHR$(134);"BY PRESSI
NG";CHR$(136);CHR$(131);" ? "
1240 PRINTTAB(0,10);CHR$(134);"YOU MOVE
YOUR SHIP UP WITH";CHR$(136);CHR$(131);
"A"
1250 PRINTTAB(0,11);CHR$(134);"AND DOWN
WITH";CHR$(136);CHR$(131);"Z"
1260 PRINTTAB(0,12);CHR$(134);"TO CHANG
E DIRECTION PRESS";CHR$(136);CHR$(131);"
'SPACE'"
1270 PRINTTAB(0,14);CHR$(134);"WHEN YOU
HAVE NEARLY RUN OUT OF CELLS,"
1280 PRINTTAB(0,15);CHR$(134);"A WARNIN
G SIREN WILL BE HEARD"
1290 PRINTTAB(0,18);CHR$(129);"PRESS SP
ACE BAR TO PLAY"
1300 REPEAT UNTIL GET$=" "
1310 CLS:PRINTTAB(0,2);CHR$(131);TAB(0,
4);CHR$(132)
1320 INPUT TAB(1,2)"INITIAL NO. OF CELL
S (5-19)",GRD:IF GRD>19 OR GRD<5 THEN 13
20

```

```

1330 INPUT TAB(1,4)"INITIAL NO. OF INVA
DERS (1-10)",SHEET:IF SHEET<1 OR SHEET>1
0 THEN 1330
1340 NUMB=SHEET:FLG=-1
1350 ENDPROC
1360 DEFPROC GAMEEND
1370 CLS
1380 PRINTTAB(0,1);CHR$(131);"YOU SCORE
D ";SCORE
1390 IF SCORE>HSCORE THEN HSCORE=SCORE:
PRINTTAB(0,3);CHR$(133);"AND THAT'S THE
HIGHEST TODAY" ELSE PRINTTAB(0,3);CHR$(1
33);"THE HIGHEST WAS ";HSCORE
1400 PRINTTAB(0,6);"PRESS SPACE BAR TO
PLAY AGAIN"
1410 REPEAT UNTIL GET$=""
1420 CLS:ENDPROC
1430 REPORT:PRINTERL
>

```


DR 2000

```
0 REM DR2000
10 HSCORE=0: ENVELOPE1,3,-4,-1,2,6,6,2
8,81,-4,-5,-1,126,63
20 MODE7
30 PROCINTRO
40 MODE 5
50 ROADX=6: WIDE=5: CARX=8: SCORE=0
60 PROCDEFINE
70 PROCROAD
80 PROCCAR
90 IF JOY THEN PROCJOY ELSE PROCMOVE
100 PROCMAN
110 PROCCAR2
120 PROCDETECT
130 GOTO70
140 DEFPROCDEFINE
150 VDU23,240,60,189,255,165,36,36,189
,255,23,241,189,36,0,0,0,0,0
160 CAR#=CHR$240
170 VDU23,242,56,56,16,56,84,84,84,16,
23,243,56,40,108,68,68,68,198,0
180 MAN#=CHR$242+CHR$10+CHR$8+CHR$243
190 VDU23,244,189,255,189,60,189,255,1
89,24,23,255,255,255,255,255,255,255
,255
200 CAR2#=CHR$244
210 CLS
220 VDU23;8202;0;0;0;0;0;
230 ENDPROC
240 DEFPROCROAD: COLOUR2
250 IFRND(100)<15 AND ROADX<18-WIDE TH
EN ROADX=ROADX+1: GOTO270
260 IFRND(100)<15 AND ROADX>1 THEN ROA
DX=ROADX-1
270 IF RND(100)<20 AND WIDE<15 THEN WI
DE=WIDE+1: GOTO290
```

```

280 IFRND(100)<20 AND WIDE>4 THEN WIDE
= WIDE-1
290 PRINTTAB(ROADX,0);CHR$(255):PRINTT
AB(ROADX+WIDE,0);CHR$(255)
300 PRINTTAB(0,0);CHR$11
310 ENDPROC
320 DEFPROCCAR:COLOUR2
330 PRINTTAB(CARX,30);CAR$
340 ENDPROC
350 DEFPROCMOVE
360 IF NOT INKEY(-98) AND NOT INKEY(-8
3) THEN ENDPROC
370 CARX=CARX+INKEY(-98)-INKEY(-83)
380 ENDPROC
390 DEFPROCMAN:COLOUR3
400 IFRND(100)<5 THENPRINTTAB(ROADX+RN
D(WIDE-1),0);MAN$
410 ENDPROC
420 DEFPROCCAR2
430 COLOUR1
440 IF RND(100)>20 THEN ENDPROC
450 PRINTTAB(ROADX+RND(WIDE-1),0);CAR2
$
460 ENDPROC
470 DEFPROCDETECT
480 X%=(CARX*64)+20:Y%=80
490 COL%=POINT(X%,Y%)
500 IF COL%=1 THEN PROCEXPLOSION
510 IF COL%=3 THEN SOUND 1,2,100,10:SC
ORE=SCORE+50
520 IF COL%=2 THEN PROCEXPLOSION
530 GCOL0,0:PLOT69,X%,Y%
540 ENDPROC
550 DEFPROCEXPLOSION
560 MOVEX%,Y%
570 FORT=1T075:SOUND0,1,100+(T/10),1:M
OVEX%,Y%-100:DRAWRND(1280),RND(1024):GCO
LO,RND(5):NEXT
580 FORT=1T01000:NEXT:CLS

```

```

590 PRINT TAB(0,0); "SCORE="; SCORE
600 IF SCORE>HSCORE THEN HSCORE=SCORE:
PRINT"THE HIGHEST TODAY":COLOUR2: ELSE P
RINT" THE HIGHEST WAS ";HSCORE
610 COLOUR2
620 PRINTTAB(0,6)"PRESS ANY KEY
TO PLAY AGAIN"
630 *FX 15 0
640 A$=GET$
650 GOTO50
660 DEFPROCTEST
670 *FX15,0
680 PRINT"ARE YOU USING JOYSTICKS?"
690 JOY=(GET$="Y")
700 DEFPROCJOY
710 CARX=CARX+(ADVAL(1)<1000)-(ADVAL(1
)>50000)
720 ENDPROC
730 DEF PROCINTRO
740 VDU 31,10,10,141
750 PRINT"Death Race 2000"
760 VDU 31,10,11,141
770 PRINT"Death Race 2000"
780 T=TIME
790 REPEAT
800 UNTIL INKEY(-99) OR TIME-T>200
810 PROCTEST
820 ENDPROC

```

>

DODGE

```

      O REM DODGE
10  MODE 7
20  PRINT TAB(10,10);CHR$(141);"Dodge"
30  PRINT TAB(10);CHR$(141);"Dodge"
40  INPUT "What level",D
50  MODE 2
60  VDU 23;8202;0;0;0;
70  DIM S(19,29)
80  COLOUR 132
90  CLS
100 VDU 23,244,170,85,170,85,170,85,17
0,85
110 VDU 23,245,0,198,254,255,255,254,1
98,0
120 VDU 23,246,0,99,127,255,255,127,99
,0
130 VDU 23,247,24,126,126,60,60,126,12
6,0
140 VDU 23,248,0,126,126,60,60,126,126
,24
150 COLOUR 1
160 FOR V=0 TO D*25
170   A=RND(20)-1
180   B=RND(30)-1
190   SOUND 1,-15,A+B*2,1
200   S(A,B)=1
210   PRINT TAB(A,B);CHR$(244)
220   NEXT V
230 S(19,29)=0
240 S(0,0)=0
250 DF=1
260 X=0
270 Y=0
280 COLOUR 5
290 PRINT TAB(19,29);"*"
300 COLOUR 3
```

```

310 TIME=0
320 SOUND 1,-15,X+Y*2,2
330 PRINT TAB(X,Y);CHR$(244+DF)
340 X1=X
350 Y1=Y
360 PROC_MOVE
370 PRINT TAB(X1,Y1);" "
380 IF X=19 AND Y=29 THEN T=TIME:MODE
4:PROC_FINISHED
390 PRINT TAB(0,30);TIME/100
400 GOTO320
410 DEF PROC_MOVE
420 IF INKEY(-83)=-1 THEN DF=1:X=X+1
430 IF INKEY(-98)=-1 THEN DF=2:X=X-1
440 IF INKEY(-73)=-1 THEN DF=3:Y=Y-1
450 IF INKEY(-105)=-1 THEN DF=4:Y=Y+1
460 IF X<0 THEN X=0
470 IF X>19 THEN X=19
480 IF Y<0 THEN Y=0
490 IF Y>29 THEN Y=29
500 IF X>=19 THEN 520
510 IF DF=1 AND S(X+1,Y)=1 THEN PROC_C
RASH
520 IF X=0 THEN 540
530 IF DF=2 AND S(X-1,Y)=1 THEN PROC_C
RASH
540 IF Y=0 THEN 560
550 IF DF=3 AND S(X,Y-1)=1 THEN PROC_C
RASH
560 IF Y>=29 THEN 580
570 IF DF=4 AND S(X,Y+1)=1 THEN PROC_C
RASH
580 ENDPROC
590 DEF PROC_CRASH
600 CLS
610 VDU 19,0,0,0,0,0
620 *FX 15,1
630 MOVE 500,400
640 DRAW 600,400

```

```

650 DRAW 600,700
660 DRAW 800,700
670 DRAW 800,800
680 DRAW 600,800
690 DRAW 600,900
700 DRAW 500,900
710 DRAW 500,800
720 DRAW 300,800
730 DRAW 300,700
740 DRAW 500,700
750 DRAW 500,400
760 PRINT TAB(6,8);"R.I.P."
770 FOR A=255 TO 100 STEP -1
780     SOUND 1,0,A,1
790     SOUND 0,-15,3,1
800     NEXT A
810 PRINT TAB(0,25);
820 GOTO 940
830 DEF PROC_FINISHED
840 *FX 15,1
850 CLS
860 VDU 19,0,0,0,0,0,0
870 PRINT TAB(0,10);"You completed a 1
level ";D;" maze"
880 PRINT TAB(0,11);"and in ";T/100;"
seconds"
890 IF T/100/D<10 THEN PRINT"That is
great" ELSE PRINT"That is rubbish"
900 FOR A=100 TO 255
910     SOUND 1,0,A,1
920     SOUND 0,-15,3,1
930     NEXTA
940 PRINT"Do you want another go?";
950 REPEAT
960     G$=GET$
970     UNTIL G$="Y" OR G$="N"
980 PRINTG$
990 IF G$="Y" THEN RUN
1000 END

```

DOWNHILL SKIING

```
      0 REM DOWNHILL SKIING
      10 MODE 7
      20 PRINT TAB(10,10);CHR$(141);"Downhi
11 skiing"
      30 PRINT TAB(10);CHR$(141);"Downhill
skiing"
      40 A=INKEY(100)
      50 MODE 5
      60 COLOUR 131
      70 CLS
      80 COLOUR 1
      90 GCOL 0,0
     100 X1=15
     110 S=100
     120 C=0
     130 M=0
     140 VDU 23,244,28,28,8,127,8,20,34,65
     150 VDU 23;8202;0;0;0;
     160 TIME=0
     170 REPEAT A=A+8
     180   X=SIN(RAD(A))
     190   SOUND 1,0,S,2.5
     200   SOUND 0,-15,7,2.5
     210   X1=X1+INKEY(-98)-INKEY(-83)
     220   IF X1<0 THEN X1=0
     230   IF X1>19 THEN X1=19
     240   MOVE 500+X*500,50
     250   DRAW 500+X*500,0
     260   MOVE 850+X*400,50
     270   DRAW 850+X*400,0
     280   PRINT TAB(X1,10);CHR$(244)
     290   PRINT TAB(0,31);" "
     300   PROC_OUT
     310   PRINT TAB(X1,9);" "
     320   S=S+1
     330   M=M+1
```

```

340 IF S>255 THEN S=100
350 IF C>200 THEN MODE 4:PROC_BROKEN
_LEG
360 UNTIL M>750
370 T=TIME
380 *FX 15
390 GOTO 460
400 DEF PROC_OUT
410 N=X1*64+16
420 FOR V=N TO N+16
430 IF POINT(V,672)=0 THEN C=C+1:SOU
ND 1,-15,0,2:FOR D=1 TO 500:NEXT D
440 NEXT V
450 ENDPROC
460 MODE 4
470 S=(T DIV 100)MOD 60
480 M=(T DIV 6000)MOD 60
490 H=T-(M*6000+S*100)
500 PRINT "You crashed ";C;" times"
510 PRINT "You took ";M;" minutes ";S;"
. ";H
520 PRINT "Another go?";
530 *FX 15
540 A$=GET$
550 IF A$<>"Y" AND A$<>"N" THEN 540
560 PRINT A$
570 IF A$="Y" THEN RUN
580 END
590 DEF PROC_BROKEN_LEG
600 VDU 19,1,1,0,0,0
610 MOVE 640,512
620 MOVE 700,512
630 PLOT 85,700,800
640 MOVE 640,512
650 MOVE 640,800
660 PLOT 85,700,800
670 MOVE 500,650
680 MOVE 850,650
690 PLOT 85,850,700

```



```
700 MOVE 500,650
710 MOVE 500,700
720 PLOT 85,850,700
730 PRINTTAB(27,10); "<-Red Cross"
740 PRINT TAB(0,15); "You crashed more
than 200 times!"
750 PRINT "Your leg is now fractured in
";30+RND(20); " places"
760 PRINT "Will you try again when your
leg is"
770 PRINT "better?"
780 GOTO 520
>
```

FRUIT MACHINE

```
      0 REM FRUIT MACHINE
      10 MODE 7
      20 PRINTTAB(10,10);CHR$(141);"Fruit M
achine"
      30 PRINTTAB(10);CHR$(141);"Fruit Mach
ine"
      40 A=INKEY(100)
      50 MODE2
      60 COLOUR 1
      70 PRINTTAB(3);"Fruit Machine"
      80 COLOUR 135
      90 VDU23,240,0,15,63,63,127,127,255,2
55
      100 VDU 23,241,128,192,248,248,252,252
,254,254
      110 VDU 23,242,255,255,127,127,63,19,0
,0
      120 VDU23,243,254,254,252,252,248,112,
0,0
      130 VDU 23,244,0,0,0,3,7,7,7,7
      140 VDU 23,245,0,0,192,128,0,0,0,0
      150 VDU 23,246,7,7,7,7,3,1,0,0
      160 VDU 23,247,0,0,0,128,192,224,248,2
4
      170 VDU 23,248,0,127,127,1,1,1,1,1
      180 VDU 23,249,0,254,254,128,128,128,1
28,128
      190 VDU 23,250,1,1,1,129,193,103,62,28
      200 VDU 23,251,128,128,128,128,128,0,0
,0
      210 VDU 23,252,0,0,3,6,6,3,1,3
      220 VDU 23,253,0,0,192,96,96,192,128,1
92
      230 VDU 23,254,7,15,31,31,63,63,63,0
      240 VDU 23,255,224,240,248,248,252,252
,252,0
```

```

250  VDU 28,4,19,18,14
260  CLS
270  DIM R(3)
280  FOR A=1 TO 3
290    R(A)=RND(4)
300  NEXT A
310  IF R(1)=R(2) THEN 280
320  IF R(2)=R(3) THEN 280
330  IF R(1)=R(3) THEN 280
340  H1=FALSE
350  H2=FALSE
360  H3=FALSE
370  REPEAT
380    IF INKEY(-98)=-1 THEN H1=TRUE
390    IF INKEY(-67)=-1 THEN H2=TRUE
400    IF INKEY(-83)=-1 THEN H3=TRUE
410    IF H1=FALSE AND R(1)<=4 THEN R(1
)=R(1)+1
420    IF H1=FALSE AND R(1)>4 THEN R(1)
=1
430    IF H2=FALSE AND R(2)<=4 THEN R(2
)=R(2)+1
440    IF H2=FALSE AND R(2)>4 THEN R(2)
=1
450    IF H3=FALSE AND R(3)<=4 THEN R(3
)=R(3)+1
460    IF H3=FALSE AND R(3)>4 THEN R(3)
=1
470    IF H1=TRUE AND H2=TRUE AND H3=TR
UE THEN MODE 2:PROC_END
480    PROC_FRUITS
490    UNTIL FALSE
500  DEF PROC_FRUITS
510    V=3
520    FOR A=1 TO 3
530      SOUND A,-15,R(A)*40,1
540      C=R(A)*4+236
550      IF C=240 THEN COLOUR 1
560      IF C=244 THEN COLOUR 3

```

```

570 IF C=248 THEN COLOUR4
580 IF C=252 THEN COLOUR 3
590 PRINTTAB(A*3,V);CHR$(C);CHR$(C+1
)
600 PRINTTAB(A*3,V+1);CHR$(C+2);CHR$
(C+3)
610 NEXTA
620 PRINTTAB(0,5)''
630 ENDPROC
640 DEF PROC_END
650 IF R(1)=R(2) AND R(2)=R(3) THEN M=
100:GOTO 700
660 IF R(1)=R(2) THEN M=50:GOTO 700
670 IF R(1)=R(3) THEN M=50:GOTO 700
680 IF R(2)=R(3) THEN M=50:GOTO 700
690 M=0
700 IF R(1)=1 OR R(2)=1 OR R(3)=1 THEN
F=1
710 IF R(1)=2 OR R(2)=2 OR R(3)=1 THEN
F=2
720 IF R(1)=3 OR R(2)=3 OR R(3)=2 THEN
F=4
730 IF R(1)=4 OR R(2)=4 OR R(3)=4 THEN
F=3
740 PROC_FRUITS
750 COLOUR7
760 PRINT"="';M*F
770 PRINT
780 PRINT"Another go?";
790 A$=GET$
800 PRINTA$;
810 IF A$<>"N" AND A$<>"Y" THEN VDU 12
7:GOTO 790
820 IF A$="Y" THEN RUN
830 PRINT
840 END
>

```

GOLD MINE

```
      0 REM GOLD MINE
      10 MODE7
      20 PRINTTAB(10,10);CHR$(141);"Gold Mi
ne"
      30 PRINTTAB(10);CHR$(141);"Gold Mine"
      40 A=INKEY(100)
      50 CLS
      60 PRINTCHR$(132);CHR$(136);CHR$(141)
;"Select Tools"
      70 PRINTCHR$(132);CHR$(136);CHR$(141)
;"Select Tools"
      80 PRINT
      90 PRINT"1.....Spade"
     100 PRINT"2.....Pick"
     110 PRINT"3.....Pump"
     120 PRINT"4.....Helmet
"
     130 PRINT"5.....Budgie
"
     140 PRINT"6.....Dynamite"
     150 PRINT"7.....Rope"
     160 PRINT
     170 PRINT"Select any four"
     180 DIMT(4),P(10,2)
     190 M=0
     200 FOR C=0 TO 10
     210   P(C,1)=RND(20)
     220   P(C,2)=RND(26)
     230   NEXTC
     240 FOR A=1 TO 4
     250   PRINT"Select tool number ";A;
     260   INPUTB
     270   IF B<1 OR B>8 THEN GOTO 250
     280   T(A)=B
     290   NEXTA
```

```

300 MODE2
310 PROCDRAW
320 PROCKEYS
330 VDU 23,240,28,28,8,127,8,20,34,65
340 VDU 23;8202;0;0;0;
350 X=6
360 Y=7
370 REPEAT
380   PROCMANON
390   C$=FNCOMMAND
400   IF C$="MOVE" THEN PROCMOVE
410   IF C$="LIFT UP" THEN PROCLIFTUP
420   IF C$="LIFT DOWN" THEN PROCLIFTD
OWN
430   IF C$="DIG" THEN PROCDIG
440   IF C$="EXPLODE" THEN PROCEXPLODE
450   IF C$="FINISH" THEN PROCFINISH
460   IF C$="RESTART" THEN PROCRESTART
470   PROCMANOFF
480   PROCHAZARD
490   UNTIL FALSE
500 DEF PROCDRAW
510   GCOL 0,1
520   PROCBOX(0,800,300,800,0,600,300,60
0)
530   PROCBOX(0,500,300,500,0,300,300,30
0)
540   PROCBOX(0,250,150,250,0,200,150,20
0)
550   PROCBOX(200,250,300,250,200,200,30
0,200)
560   PROCBOX(0,100,600,100,0,50,600,50)
570   PROCBOX(600,800,1279,800,600,790,1
279,790)
580   PROCBOX(600,650,1279,650,600,450,1
279,450)
590   PROCBOX(600,250,1279,250,600,200,1
279,200)
600   PROCBOX(600,150,1279,150,600,50,12
79,50)

```

```

610 PROCSKY
620 PROCSUN
630 PROCCLOUD(200,900)
640 PROCCLOUD(1050,930)
650 PROCLIFT
660 ENDPROC
670 DEF PROCBOX(X1,Y1,X2,Y2,X3,Y3,X4,Y
4)
680 MOVE X1,Y1
690 MOVE X2,Y2
700 PLOT 85,X4,Y4
710 MOVE X3,Y3
720 PLOT 85,X1,Y1
730 PLOT85,X1,Y1
740 ENDFPROC
750 DEF PROCSKY
760 VDU 24,0;800;1279;1023;
770 GCOL 0,132
780 CLG
790 VDU 24,0;0;1279;1023;
800 GCOL0,128
810 ENDFPROC
820 DEF PROCSUN
830 GCOL 0,3
840 VDU 29,1180;900;
850 FOR A=0 TO 360 STEP 20
860   X1=SIN(RAD(A))
870   Y1=COS(RAD(A))
880   X2=SIN(RAD(A+20))
890   Y2=COS(RAD(A+20))
900   MOVE 0,0
910   MOVE X1*50,Y1*50
920   PLOT 85,X2*50,Y2*50
930   IF A/60=INT(A/60) THEN DRAW X2*80
,Y2*80
940   NEXTA
950 ENDFPROC
960 DEF PROCCLOUD(X,Y)
970 GCOL 0,7

```

```

980 VDU 29,X;Y;
990 FOR A=0 TO 360 STEP 20
1000   X1=SIN(RAD(A))*100
1010   Y1=COS(RAD(A))*50
1020   X2=SIN(RAD(A+20))*100
1030   Y2=COS(RAD(A+20))*50
1040   MOVE 0,0
1050   MOVE X1,Y1
1060   PLOT 85,X2,Y2
1070   MOVE X1,Y1
1080   X3=SIN(RAD(A+10))*110
1090   Y3=COS(RAD(A+10))*60
1100   PLOT 85,X3,Y3
1110   NEXTA
1120 ENDPROC
1130 DEF PROCLIFT
1140 GCOL 0,0
1150 VDU 29,0;0;
1160 MOVE 300,800
1170 DRAW 450,900
1180 DRAW 600,800
1190 VDU 29,450;900;
1200 MOVE0,0
1210 FOR A=0 TO 360 STEP 20
1220   X=SIN(RAD(A))*100
1230   Y=COS(RAD(A))*100
1240   DRAW X,Y
1250   IF A/20=INT(A/20) THEN MOVE 0,0;
DRAW X,Y
1260   NEXTA
1270 ENDPROC
1280 DEF PROCKEYS
1290 *KEY 0 MOVEIKIM
1300 *KEY 1 LIFT UP!KIM
1310 *KEY 2 LIFT DOWN!KIM
1320 *KEY 3 EXPLODE!KIM
1330 *KEY 4 DIG!KIM
1340 *KEY 5 FINISH!KIM
1350 *KEY 6 RESTART!KIM

```



```

1360 ENDPROC
1370 DEF FNCOMMAND
1380 PRINTTAB(0,31);"*";
1390 REPEAT *FX 15,1
1400   UNTIL INKEY(-33)=-1 OR INKEY(-11
4)=-1 OR INKEY(-115)=-1 OR INKEY(-116)=-
1 OR INKEY(-21)=-1 OR INKEY(-117)=-1 OR
INKEY(-118)=-1
1410 INPUT"B$
1420 T=TIME:REPEAT UNTIL TIME-T>100
1430 PRINTTAB(0,31);"          ";
1440 =B$
1450 DEF PROCMANON
1460 PRINTTAB(X,Y);CHR$(240)
1470 X1=X
1480 Y1=Y
1490 ENDPROC
1500 DEF PROCMANOFF
1510 PRINTTAB(X1,Y1);" "
1520 ENDPROC
1530 DEF PROCMOVE
1540 PRINTTAB(0,31);"L or R ";
1550 D$=GET$
1560 PRINTD$;
1570 T=TIME
1580 REPEAT UNTIL TIME-T>150
1590 IF D$<>"L" AND D$<>"R" THEN PRINTT
AB(0,31);"          ";;GOTO 1540
1600 IF D$="R" THEN X=X+1
1610 IF D$="L" THEN X=X-1
1620 PRINTTAB(0,31);"          ";
1630 ENDPROC
1640 DEF PROCLIFTUP
1650 GCOL 0,7
1660 VDU 29,0;0;
1670 IF X<4 OR X>8 THEN ENDPROC
1680 PRINTTAB(0,31);"Press RETURN=stop"
;
1690 C=(31-Y)*32-1

```

```

1700 MOVE 300,C: DRAW 600,C
1710 PROCMANON
1720 T=TIME: REPEAT UNTIL TIME-T=100
1730 IF INKEY(-74)=-1 OR Y<7 THEN PRINT
TAB(0,31); " ";;ENDPROC
1740 PROCMANOFF: GCOL 0,0: MOVE 300,C: DRAW
600,C: GCOL 0,7
1750 Y=Y-1
1760 C=C+32
1770 GOTO1700
1780 DEF PROCLIFTDOWN
1790 GCOL 0,7
1800 VDU 29,0;0;
1810 IF X<4 OR X>8 THEN ENDPROC
1820 PRINTTAB(0,31); "Press RETURN=stop"
;
1830 C=(31-Y)*32-1
1840 MOVE 300,C: DRAW 600,C
1850 PROCMANON
1860 IF INKEY(-74)=-1 OR Y=26 THEN PRIN
TTAB(0,31); " ";;ENDPROC
1870 T=TIME: REPEAT UNTIL TIME-T=100
1880 GCOL 0,0: MOVE 300,C: DRAW 600,C: GCOL
0,7
1890 Y=Y+1
1900 C=C-32
1910 PROCMANOFF
1920 GOTO1840
1930 DEF PROCDIG
1940 FOR C=0 TO 10
1950 IF (P(C,1)-X)<2 AND (P(C,2)-Y)<2
THEN 1980
1960 NEXTC
1970 ENDPROC
1980 G=RND(10)-1
1990 PRINTTAB(0,31); G; "lbs. of gold fou
nd";
2000 T=TIME: REPEAT UNTIL TIME-T>200

```

```

2010 M=M+G
2020 PRINTTAB(0,31);"
";
2030 P(C,1)=0:P(C,1)=0
2040 ENDPROC
2050 DEF PROCEXPLODE
2060 FOR C=0 TO 10
2070 IF (P(C,1)-X)<RND(5) AND (P(C,2)
- Y)<RND(5) THEN 1980
2080 NEXTC
2090 ENDPROC
2100 DEF PROCFINISH
2110 T=TIME:REPEAT UNTIL TIME-T>200
2120 VDU22,7
2130 PRINT"You have discovered ";M;" lb
s of"
2140 PRINT"gold. Do you want to try to
beat your"
2150 PRINT;M;" pounds?";
2160 INPUT G$
2170 IF LEFT$(G$,1)="Y" THEN RUN
2180 END
2190 DEF PROCRESTART
2200 RUN
2210 DEF PROCHAZARD
2220 H=RND(20)
2230 IF H=1 AND FNQ(3)=0 THEN PROCFLOOD
2240 IF H=2 AND FNQ(1)=0 AND FNQ(2)=0 A
ND FNQ(4)=0 AND FNQ(7)=0 THEN PROCROCK
2250 IF H=3 AND FNQ(3)=0 THEN PROCEXP
2260 ENDPROC
2270 DEF PROCFLOOD
2280 DEF PROCEXP
2290 CLS
2300 PRINTTAB(0,16);"Boom!!!!!!!!!"
2310 PROCFINISH
2320 PROCFINISH
2330 DEF PROCROCK
2340 CLS

```

```
2350 PRINT"You have been killed"
2360 PRINT"in a rockfall"
2370 PROCFINISH
2380 DEF FNO(W)
2385 N=0
2390 V=0:REPEAT V=V+1
2400   IF T(V)=W THEN N=1
2410   UNTIL V=4
2420 =N
>
```

GUESS

```
0 REM GUESS
10 MODE 7
20 PROCIN
30 PROCGM
40 PROCT
50 PROCGY
60 PROCGL
70 PROCW
80 IF A$="Y" THEN CLS:RUN
90 CLS
100 *FX15,0
110 END
120 DEF PROCIN
130 VDU 31,10,10,141
140 PRINT"Guess"
150 VDU 31,10,11,141
160 PRINT"Guess"
170 T=TIME
180 REPEAT UNTIL INKEY(-99) OR TIME-T>
200
190 CLS
200 ENDPROC
210 DEF PROCGM
220 Z=0
230 VDU 10,10,135,157,129
240 PRINT"Guess my number...";
250 VDU 156,10,13,10,10,156,131,136
260 PRINT "It is between 1 and 10000"

270 Z=Z+1
280 N=RND(10000)
290 REPEAT
300     VDU 132,157,135
310     INPUT "Guess...",G
320     Z=Z+1
330     IF N>G THEN PROCHI
```

```

340   IF N<G THEN PROCLO
350   UNTIL G=N
360 PROC0
370 PRINT "You did it in ";Z'''
380 PRINT"Just listen to that applause
!"
390 SOUND 0,-15,4,100-Z
400 T=TIME
410 REPEAT
420   UNTIL TIME-T>500-Z*5
430 ENDPROC
440 DEF PROCBY
450 CLS
460 MS=0
470 MIN=0
480 MAX=10000
490 G=RND(10000)
500 REPEAT
510   PRINT"Is it ";G;"?"
520   T=0
530   MS=MS+1
540   *FX15,1
550   A$=GET$
560   IF A$="H" THEN PROCH
570   IF A$="L" THEN PROCL
580   IF A$<>"L" AND A$<>"H" AND A$<>"
" THEN PRINT "PRESS 'H' IF I SHOULD GO
HIGHER OR          'L' IF I SHOULD GO
LOWER              ('SPACE BAR' WHEN I
GET IT RIGHT)":GOTO 550
590   UNTIL A$=" "
600 ENDPROC
610 DEF PROCH
620 PROCHI
630 MIN=G
640 G=INT(G+G/2)
650 T=T+1
660 IF T>20 THEN PROCR
670 IF T>5 AND T<20 THEN PROCS

```

```

680 IF G>MIN AND G<MAX THEN ENDPROC
690 IF G<=MIN THEN 640
700 IF G>=MAX THEN 740
710 DEF PROCL
720 PROCLO
730 MAX=G
740 G=INT(G-G/2)
750 T=T+1
760 GOTO 670
770 DEF PROCS
780 IF G>MIN AND A$="L" THEN G=G-1
790 IF G<MAX AND A$="H" THEN G=G+1
800 ENDPROC
810 DEF PROCR
820 IF A$="H" OR G>=MAX THEN G=MIN+RND
(20)
830 IF A$="L" OR G<=MIN THEN G=MAX-RND
(20)
840 ENDPROC
850 DEF PROCW
860 CLS
870 PRINT"I did it in ";MS
880 VDU 129
890 PRINTSTRING$(MS,"*")
900 PRINT"You did it in ";Z
910 VDU 130
920 PRINTSTRING$(Z,"*")
930 IF MS>Z THEN PRINT "Too bad, you b
eat me.."
940 IF MS<Z THEN PRINT "But then you'r
e only human..."
950 IF MS=Z THEN PRINT "I always say t
hat computers are human somewhere insi
de them."
960 *FX15,1
970 PRINT'"ANOTHER GO?"
980 REPEAT
990 A$=GET$

```

```

1000  UNTIL A$="Y" OR A$="N"
1010  ENDPROC
1020  DEF PROCHI
1030  SOUND 1,-10,200,10
1040  VDU 141,135
1050  PRINT"HIGHER"
1060  VDU 141,129
1070  PRINT"HIGHER"
1080  ENDPROC
1090  DEF PROCLO
1100  SOUND 2,-10,50,10
1110  VDU 141,134
1120  PRINT"LOWER"
1130  VDU 141,131
1140  PRINT"LOWER"
1150  ENDPROC
1160  DEF PROCO
1170  VDU 10,10,10,10,141,130
1180  PRINT"Corrrrrect!"
1190  VDU 141,133
1200  PRINT"Corrrrrect!"
1210  ENDPROC
1220  DEF PROCT
1230  CLS
1240  VDU 133
1250  PRINT"Now it's my turn to guess yo
ur number"
1260  VDU 10,13,10,10,131
1270  PRINT"But first you must think of
a number"
1280  VDU 10,13,10,130
1290  PRINT"Between 1 and 10000"
1300  VDU 31,0,22,132,157,135
1310  PRINT"Hit the space-bar when you'r
e ready"
1320  REPEAT
1330  UNTIL INKEY(-99)
1340  ENDPROC

```



```
1350 DEF PROCGL
1360 FOR A%=1 TO 100
1370   SOUND RND(4)-1,-10,RND(255),1
1380   NEXT
1390 ENDPROC
>
```

HOP IT!!

```
0 REM HOP IT!
10 MODE 7
20 PRINTTAB(10,10);CHR$(136);CHR$(141
);"Hop it!"
30 PRINTTAB(10);CHR$(136);CHR$(141);"
Hop it!"
40 VDU 23,244,255,255,255,255,255,255
,255,255
50 A=INKEY(200)
60 N=0
70 S=0
80 MODE 2:VDU 23;8202;0;0;0;0;
90 PROC_INIT
100 COLOUR 1
110 PROC_DRAW
120 VDU 23,245,28,28,8,127,8,20,34,65

130 J=0
140 BX=0
150 BY=4
160 F=0
170 COLOUR 3
180 PRINT TAB(BX,BY);"O"
190 COLOUR 2
200 PRINT TAB(16,20+J);CHR$(245)
210 J1=J
220 B1X=BX
230 B1Y=BY
240 V=RND(3)-1
250 IF BY<20 THEN BY=BY+V
260 BX=BX+V
270 SOUND 1,-15,200-BY*10,1
280 COLOUR 4:COLOUR 130
290 PRINT TAB(0,25);"Score ";S
300 PROC_MEN
310 COLOUR 128
```

```

320 IF INKEY(-70)=-1 AND J>-5 AND F=0
THEN J=J-1
330 IF J=-5 THEN F=1:J=J+1
340 IF J=0 THEN F=0
350 IF F=1 THEN J=J+1
360 IF INKEY(-70)=0 AND J<>0 THEN J=J+
1
370 PRINT TAB(16,20+J1);" "
380 PRINT TAB(B1X,B1Y);" "
390 IF BX>16 THEN 410
400 GOTO170
410 IF J<-2 THEN S=S+10:GOTO 130
420 N=N+1
430 SOUND 1,-15,0,10
440 PROC_OUCH
450 IF N<6 THEN GOTO 130
460 COLOUR 128:COLOUR 5
470 CLS
480 PRINT "Final Score ";S
490 PRINT"Another go?"
500 G#=GET#
510 IF G#<>"Y" AND G#<>"N" THEN 500
520 IF G#="Y" THEN RUN
530 COLOUR 128
540 COLOUR 7
550 CLS
560 END
570 DEF PROC_OUCH COLOUR 3
580 PRINT TAB(16,20);CHR$(245);"OW!"
590 FOR A=1 TO 20
600     PRINT TAB(17,20);"OW!"
610     FOR D=1 TO 100
620         NEXT D
630     PRINT TAB(17,20);"    "
640     FOR D=1 TO 100
650         NEXT D
660     NEXT A
670 ENDPROC
680 DEF PROC_MEN

```

```

690 H=5-N
700 IF H=1 THEN ENDPROC
710 PRINT TAB(0,30);
720 FOR X=1 TO H
730   PRINTCHR$(245);
740   NEXTX
750 FOR A=H+1 TO 20
760   PRINT" ";
770   NEXTA
780 ENDPROC
790 DEF PROC_DRAW
800 FOR A=0 TO 15
810   FOR B=A+5 TO 20
820     PRINT TAB(A,B);CHR$(244)
830     NEXTB,A
840   ENDPROC
850   DEF PROC_INIT
860   COLOUR 4
870   FOR A=0 TO 3
880     FOR B=0 TO 19
890       PRINTTAB(B,A);CHR$(244)
900       NEXT B,A
910     COLOUR 2
920     FOR A=21 TO 29
930       FOR B=0 TO 19
940         PRINTTAB(B,A);CHR$(244)

950         NEXT B,A
960       ENDPROC

```

>

INVADERS

```

    0 REM INVADERS
    10 MODE 7:PROCINTRO
    20 MODE5:SC=0:LV=3:PX=10:FL=-1:HS=0
    30 PROCSETUP
    40 PROCSCREEN:SC=0
    50 PROCGAME
    60 END
    70 DEFPROCSETUP
    80 VDU19,0,4;0;
    90 VDU23;8202;0;0;0;
    100 VDU23,240,24,60,126,219,90,126,66,
129
    110 INV#=CHR$(240)
    120 VDU23,241,0,0,16,16,255,255,255,25
5
    130 BASE#=CHR$(241)
    140 ENDPROC
    150 DEFPROCSCREEN
    160 FOR T=1 TO 40:COLOUR2
    170   PRINTTAB(RND(16)+1,RND(20));INV#
    180   NEXT:COLOUR1
    190 PRINTTAB(0,0)"
"
    200 PRINTTAB(0,0)"SCORE-";SC
    210 ENDPROC
    220 DEFPROCGAME
    230 REPEAT
    240   IF RND(100)<LEVEL THEN PROCMOV:P
ROCCHK
    250   PROCMOVE
    260   PROCUPDATE
    270   UNTIL FL=0
    280 CLS
    290 PRINTTAB(0,0);"YOU SCORED-";SC
    300 IF SC>HS THEN HS=SC:PRINT"THE HIGH
EST TODAY" ELSE PRINT "THE HIGEST WAS ";H
S

```

```

310 *FX15 0
320 PRINT:PRINT"PRESS ANY KEY "
330 A#=GET#:CLS:FL=-1
340 GOTO40
350 DEFPROCMOV
360 PRINTTAB(0,0); "
":PRINTTAB(PX,30); " "
370 COLOUR2:FORT=1TO3:PRINTTAB(RND(16)
+1,0);INV#:NEXT:COLOUR1
380 PRINTTAB(0,0);CHR#11;CHR#11
390 PRINTTAB(0,0);"SCORE-";SC
400 ENDPROC
410 DEFPROCMOVE
420 IF INKEY(-98) THEN PX=PX-0.1
430 IF INKEY(-83) THEN PX=PX+0.1
440 IFPX>16 THEN PX=16
450 IF INKEY(-74) THEN PROCFIRE
460 COLOUR1:PRINTTAB(PX,30); " ";BASE#;
" ":COLOUR3
470 ENDPROC
480 DEFPROCFIRE
490 IFRND(100)<4 THEN PROCMOVE
500 FORT=29 TO 0 STEP-1
510 IF POINT((INTPX*64+100),ABS(1024
-(T*32-40)))=2 THEN U=T:T=-1:H=-1
520 GCOLOR,7
530 PLOT69,(INTPX*64+100),ABS(1024-(
T*32-40))
540 GCOLOR,4
550 PLOT69,(INTPX*64+100),ABS(1024-(
(T+1)*32-40))
560 NEXT
570 IF NOT H THEN ENDPROC
580 SOUND0,-15,100,5
590 SC=SC+RND(5)*10
600 PRINTTAB(INTPX+1,U-1); " ";TAB(INT
PX+1,U-2); " "
610 H=0
620 ENDPROC

```

```

630 DEFPROCUPDATE
640 PRINTTAB(0,0); "SCORE-"; SC
650 ENDPROC
660 DEFPROC CHECK
670 FORX=1TO1280STEP10
680   IF POINT(X,20)=2 THEN FL=0
690   NEXTX
700 ENDPROC
710 DEFPROCLEVEL
720   PRINT"LEVEL(2-100)""'2-EASY 100 I
MPOS."
730 REPEAT
740   LEVEL=VALGET$
750   UNTIL LEVEL>1 AND LEVEL<101
760 ENDPROC
770 DEF PROCINTRO
780 VDU 31,10,10,141
790 PRINT"Invaders"
800 VDU 31,10,11,141
810 PRINT"Invaders"
820 T=TIME
830 REPEAT
840   UNTIL INKEY(-99) OR TIME-T>200
850 PROCLEVEL
860 ENDPROC

```

>

LANDER

```
0 REM LANDER
10 MODE 7
20 PROCIN
30 PROCVA
40 PROCUS
50 MODE 2
60 VDU 5
70 PROCBD
80 PROCPS
90 PROCPF
100 REPEAT
110   PROCPS
120   U%=U%+UV%/10
130   P%=P%+PV%/10
140   PROCPS
150   SOUND 1,-7,F%/50,2
160   SOUND 2,-10,ABSUV%/3,5
170   SOUND 3,-10,ABSPV%/3,5
180   UV%=UV%+UA%
190   PV%=PV%+PA%
200   UV%=UV%-SGN(UV%)
210   IF F%<0 THEN 270
220   UA%=UA%+INKEY(-98)-INKEY(-83)
230   IF NOT(INKEY(-98) OR INKEY(-83))
THEN UA%=0
240   IF NOT INKEY(-73) THEN PA%=-10
250   PA%=PA%-INKEY(-73)*2
260   PROCPF
270   F%=F%-ABS(UA%)-ABS(PA%)
280   IF PA%=-10 THEN F%=F%+10
290   PROCPF
300   UNTIL P%<=I%+48
310 IF ABS(U%-J%)>64 OR PV%<-320 THEN
PROCEX ELSE PROCGD
320 PROCNG
330 END
```



```

340 DEF PROCIN
350 VDU 31,10,10,141
360 PRINT"Lander"
370 VDU 31,10,11,141
380 PRINT"Lander"
390 T=TIME
400 REPEAT
410   UNTIL INKEY(-99) OR TIME-T>200
420 ENDPROC
430 DEF PROCVA
440   U%=RND(956)
450   P%=1023
460   PA%=-10
470   UA%=0
480   PV%=0
490   UV%=0
500   F%=5000
510 ENDPROC
520 DEF PROCUS
530 VDU 23,255,7,15,9,9,31,35,64,240
540 VDU 23,254,224,240,144,144,248,196
,2,15
550 VDU 23,253,0,8,4,126,4,8,0,0
560 VDU 23,252,0,0,16,32,126,32,16,0
570 VDU 23,251,0,16,56,84,16,16,16,0
580 VDU 23,250,0,8,8,8,42,28,8,0
590 ENDPROC
600 DEF PROCPS
610 GCOL 3,6
620 MOVE U%,P%
630 VDU 255,254
640 MOVE 0,968
650 IF SGN(UV%)=-1 THEN PRINT CHR$252;
-UV%
660 IF SGN(UV%)=1 THEN PRINT CHR$253;U
V%
670 VDU 10
680 IF SGN(PV%)=-1 THEN PRINT CHR$250;
-PV%

```

```

690 IF SGN(PV%)=1 THEN PRINT CHR$251;P
V%
700 ENDPROC
710 DEF PROCBD
720 FOR A%=1 TO 200
730     GCOL 0,RND(7)
740     PLOT 69,RND(1279),RND(1023)
750     NEXT
760 L%=0
770 MOVE 0,0
780 I%=0
790 FOR A%=0 TO 1279 STEP 32
800     DRAW A%,L%
810     L%=L%+RND(64)-RND(64)
820     IF I%<L% THEN I%=L%:J%=A%
830     IF L%<0 THEN L%=0
840     NEXT
850 GCOL 0,9
860 IF J%>1151 THEN J%=1151
870 MOVE J%,I%
880 MOVE J%+128,I%
890 PLOT 85,J%+128,I%+16
900 MOVE J%,I%
910 PLOT 85,J%,I%+16
920 ENDPROC
930 DEF PROC EX
940 SOUND 0,-15,4,-1
950 FOR A%=1 TO 100
960     GCOL 0,RND(7)
970     DRAW RND(1279),RND(1023)
980     NEXT
990 *FX15,0
1000 ENDPROC
1010 DEF PROC GD
1020 PROCPS
1030 P%=I%+48
1040 U%=J%
1050 PROCPS
1060 GCOL 3,7

```

```

1070 MOVE 452,800
1080 PRINT"Well Done"
1090 MOVE 452,768
1100 PRINT"*****"
1110 MOVE 452,512
1120 PRINT"Fuel Left"
1130 MOVE 640-(LEN(STR$F%)*64)/2,470
1140 PRINT ;F%
1150 ENDPROC
1160 DEF PROCNG
1170 *FX15,0
1180 REPEAT
1190   UNTIL INKEY(-69) OR INKEY(-86)
1200 IF INKEY(-69) THEN RUN
1210 CLS
1220 ENDPROC
1230 DEF PROC PF
1240 MOVE 0,1000
1250 GCOL 3,1
1260 PRINT ;F%
1270 ENDPROC

```

>

MORSE

```

      0 REM MORSE
      10 MODE7
      20 PRINTTAB(10,10);CHR$(141);"Morse C
ode"
      30 PRINTTAB(10);CHR$(141);"Morse Code
"
      40 A=INKEY(100)
      50 CLS
      60 DIM M$(100),A$(36)
      70 FOR A=1 TO 36
      80   READ A$(A)
      90   NEXT A
     100 DATA .-
     110 DATA -...
     120 DATA -.-.
     130 DATA -..
     140 DATA .
     150 DATA ..-.
     160 DATA --.
     170 DATA ....
     180 DATA ..
     190 DATA .---
     200 DATA -.-
     210 DATA .-..
     220 DATA --
     230 DATA -.
     240 DATA ---
     250 DATA .--.
     260 DATA --.-
     270 DATA .-.
     280 DATA ...
     290 DATA -
     300 DATA -.-
     310 DATA ...-
     320 DATA .--
     330 DATA -.-.-
```

```

340 DATA -.--
350 DATA ---..
360 DATA .-----
370 DATA ..----
380 DATA ...--
390 DATA ....-
400 DATA .....
410 DATA -....
420 DATA ---...
430 DATA ----..
440 DATA -----
450 DATA -----
460 FOR A=1 TO 100
470   READ M$(A)
480   NEXTA
490 DATA BBC COMPUTER 32K,BASIC,MACHIN
E CODE,PASCAL,FORTH,MEMORY,MICROPROCESSO
R,PRINTER,FORTRAN,COMPUTERS ARE GREAT,RA
M,ROM,SOS,MORSE CODE,HELLO,CASSETTE,DISC
DRIVE,GRAPHICS,COLOUR,SOUND,ECONET,ANAL
OGUE,DIGITAL,SPACE INVADERS,PROGRAM
500 DATA INTERFACE,ADDRESS,DATA,OPERAT
ING SYSTEM,MOS,EPROM,DOS,CPU,MONITOR,ASS
EMBLER,6502,280,8080,WORD PROCESSOR,WELC
OME,ERROR,BUG,SOFTWARE,HARDWARE
510 DATA TEST,ARRAY,BIT,BYTE,CURSOR,DA
TA,HARD COPY,HEX,HIGH RESOLUTION,LOOP,MO
DULATOR,PEEK,POKE,PORT
520 DATA SUBROUTINE,REGISTER,TAB,TOOLK
IT,VDU,SYNTAX,STRING
530 DATA ZX81,VIC,SPECTRUM,ATOM,DRAGON
,ATARI,VIDEO GENIE,TRS80,TEXAS TI994A,SH
ARP,NASCOM,DAI,PET,APPLE,JUPITER,NEWBRAI
N,ORIC,INSTRUCTION,MASK,BINARY,PACKMAN
540 DATAMICRO,MINI,KEY,KEYBOARD,FUNCTI
ON,ADVENTURE,INPUT,OUTPUT,MAINFRAME,PRIN
TOUT,BUFFER,LEAD,PLUG,SYSTEM
550 REPEAT
560   C$=M$(RND(100))

```

```

570   FOR A=1 TO LEN C$
580     PROCSOUND(MID$(C$,A,1))
590     SOUND 1,0,1,5
600     NEXTA
610     INPUT "What was that phrase",P$
620     IF P$=C$ THEN PRINT "Well done. T
hat is correct"
630     IF P$<>C$ THEN PRINT "Do you wish
to hear it again?":G$=GET$:IF G$="Y" TH
EN 570
640     IF P$<>C$ THEN PRINT "It was ";C$
650     PRINT "-----"
-----"
660     PRINT "PRESS ANY KEY WHEN READY"
670     Z$=GET$
680     UNTIL FALSE
690 DEF PROCSOUND(S$)
700 RESTORE 820
710 Z=-1:REPEAT Z=Z+1
720   READ N$,N
730   UNTIL N$=S$ OR Z=35
740 IF S$=" " THEN SOUND 1,0,1,7:ENDPR
OC
750 D$=A$(N)
760 FOR K=1 TO LEN(D$)
770   IF MID$(D$,K,1)="." THEN SOUND 1
,-15,53,2
780   IF MID$(D$,K,1)="-" THEN SOUND 1
,-15,53,4
790   SOUND 1,0,53,3
800   NEXTK
810 ENDPROC
820 DATA A,1,B,2,C,3,D,4,E,5,F,6,G,7,H
,B,I,10,J,11,K,12,L,12,M,13,N,14,O,15,P,
16,Q,17,R,18,S,19,T,20,U,21,V,22,W,23,X,
24,Y,25,Z,26,1,27,2,28,3,29,4,30,5,31,6,
32,7,33,8,34,9,35,0,36
>

```

NIGHTFALL

```
0 REM NIGHTFALL
10 MODE 7
20 PROCIN
30 MODE 2
40 VDU 5
50 PROCVA
60 PROCH
70 REPEAT
80   PROCBD
90   PROCPS
100  SOUND 0,-8,2,-1
110  REPEAT
120    PROCPS
130    PROCMS
140    PROCPS
150    IF INKEY(-74) THEN PROCB
160    IF F% THEN PROCMB
170    UNTIL T% OR K%=20
180  IF T% THEN PROCEX:RUN
190  PROCNL
200  UNTIL FALSE
210 END
220 DEF PROCIN
230  G%=0
240  VDU 31,10,10,141
250  PRINT"Nightfall"
260  VDU 31,10,11,141
270  PRINT"Nightfall"
280  T=TIME
290  REPEAT
300    UNTIL INKEY(-99) OR TIME-T>200
310  ENDPROC
320 DEF PROCVA
330  U%=0
340  P%=960
350  IF G% THEN 430
```

```

360 S%=0
370 F%=0
380 D%=16
390 T%=0
400 V%=1
410 *FX9,5
420 *FX10,5
430 K%=0
440 G%=-1
450 ENVELOPE 1,1,-200,-200,-200,1,2,3,
120,-20,0,-20,120,80
460 ENVELOPE 2,2,-1,0,0,100,0,0,90,0,0
,-90,90,90
470 ENVELOPE 3,1,0,0,0,0,0,0,127,-1,-1
,-1,127,0
480 ENDPROC
490 DEF PROCH
500 DIM C$(9)
510 FOR A%=0 TO 9
520   C$(A%)=CHR$18+CHR$3+CHR$A%
530   NEXT
540 VDU 23,255,255,153,153,255,255,153
,153,255
550 VDU 23,254,0,102,102,0,0,102,102,0
560 VDU 23,253,0,0,192,255,127,28,56,1
6
570 VDU 23,252,0,0,14,0,0,0,0,0
580 S%=C$(7)+CHR$253+CHR$8+C$(9)+CHR$2
52
590 VDU 23,251,0,0,0,0,0,32,28,32
600 VDU 23,250,0,128,112,128,0,0,0,0
610 ENDPROC
620 DEF PROCBD
630 FOR A%=0 TO 1279 STEP 64
640   FOR B%=32 TO RND(20+V%)*32 STEP
32
650     MOVE A%,B%
660     PRINTC$(RND(7))+CHR$255+CHR$8+
C$(RND(8)-1)+CHR$254

```



```

670     NEXT
680     NEXT
690 GCOL 3,11
700 MOVE 0,1023
710 PRINT"Score ";S%
720 MOVE 838,1023
730 PRINT"Level ";V%
740 ENDPROC
750 DEF PROCPS
760 MOVE U%,P%
770 PRINTS$
780 ENDPROC
790 DEF PROCMS
800 U%=U%+D%
810 IF U%>1279 THEN U%=0:P%=P%-32:SOUN
D 1,1,P%/10,8
820 T%=(POINT(U%,P%)<>0)
830 ENDPROC
840 DEF PROC EX
850 *FX9,1
860 *FX10,1
870 *FX15,0
880 SOUND 0,-15,4,20
890 VDU 19,0,9,0,0,0
900 T=TIME
910 REPEAT
920     UNTIL TIME-T>100
930 VDU 19,0,0,0,0,0
940 VDU 4
950 PRINT TAB(5,30);"Hit any key"
960 PRINT TAB(4,31);"to play again";
970 G=GET
980 ENDPROC
990 DEF PROCB
1000 IF F% THEN ENDPROC
1010 SOUND 18,2,200,-1
1020 F%=-1
1030 B%=(U% DIV 64)*64
1040 M%=P%

```

```

1050 PROCDB
1060 PROCPB
1070 ENDPROC
1080 DEF PROCMB
1090 PROCPB
1100 M%=M%-32
1110 IF POINT(B%+32,M%)<>0 THEN 1150
1120 IF M%<=30 THEN 1260
1130 PROCPB
1140 ENDPROC
1150 REM BB
1160 PROCSZ
1170 GCOL 0,0
1180 MOVE B%,M%
1190 MOVE B%+63,M%
1200 PLOT 85,B%,0
1210 PLOT 85,B%+63,0
1220 K%=K%+1
1230 PROCS
1240 S%=S%+M% DIV 10
1250 PROCS
1260 SOUND 18,0,0,1
1270 F%=0
1280 ENDPROC
1290 DEF PROCPB
1300 MOVE B%,M%
1310 PRINT B$
1320 ENDPROC
1330 DEF PROCDB
1340 B$=C$(RND(7))+CHR$251+CHR$B+C$(RND
(7))+CHR$250
1350 ENDPROC
1360 DEF PROCNL
1370 PROCFF
1380 FOR A%=1 TO 7
1390   VDU 19,7,A%,0,0,0
1400   T=TIME
1410   REPEAT
1420     UNTIL TIME-T>100

```

```

1430     NEXT
1440 PROCVA
1450 V%=V%+1
1460 D%=D%-2*(D%<60)
1470 S%=S%+P%
1480 CLG
1490 ENDPROC
1500 DEF PROCS
1510 GCOL 3,11
1520 MOVE 384,1023
1530 PRINT ;S%
1540 ENDPROC
1550 DEF PROCSZ
1560 SOUND 16,3,6,10
1570 SOUND 4096,-8,2,-1
1580 ENDPROC
1590 DEF PROCCF
1600 SOUND 2,0,0,40
1610 SOUND 3,0,0,80
1620 RESTORE 1720
1630 FOR A%=1 TO 4
1640     READ N%
1650     SOUND 1,-11,N%,10
1660     SOUND 2,-11,N%,10
1670     SOUND 3,-11,N%,10
1680     NEXT
1690     SOUND &102,-11,101,20
1700     SOUND &103,-11,117,20
1710 ENDPROC
1720 DATA 101,97,89,97

```

>

PARACHUTE

```
      0 REM PARACHUTE
    10 MODE7
    20 PRINTTAB(10,10);CHR$(141);"Parachu
te"
    30 PRINTTAB(10);CHR$(141);"Parachute"
    40 INPUT "What is the wind force toda
y",W1
    50 A=INKEY(100)
    60 MODE2
    70 VDU 23,240,28,28,8,127,8,20,34,65
    80 VDU 23,241,0,24,60,126,255,66,36,2
4,16
    90 VDU 23,242,255,255,255,255,255,199
,199,199
   100 VDU 23,243,0,0,128,192,252,230,252
,0
   110 VDU 23;8202;0;0;0;
   120 DIM S(19,31)
   130 COLOUR 2
   140 FOR X=0 TO 19 STEP 3
   150   FOR Y=30 TO 30-RND(10) STEP -1
   160     PRINTTAB(X,Y);CHR$(242)
   170     S(X,Y)=1
   180     NEXT Y,X
   190   PROCAIROPLANE
   200   X=10
   210   Y=0
   220   W=RND(W1/2)
   230   IF RND(2)=1 THEN W=-W
   240   W=W+INKEY(-98)-INKEY(-83)
   250   Y=Y+1
   260   X=X+W
   270   IF X>19 THEN X=19
   280   IF X<0 THEN X=0
```

```

290 IF Y=30 THEN PROC GOOD
300 SOUND 1,-15,Y*20,1
310 SOUND 2,-15,255-Y*20,1
320 COLOUR 1
330 PRINTTAB(X,Y);CHR$(241)
340 COLOUR 3
350 PRINTTAB(X,Y+1);CHR$(240)
360 T=TIME:REPEAT UNTIL TIME-T>10
370 PRINTTAB(X,Y);" ";TAB(X,Y+1);" "
380 IF S(X,Y)=1 THEN PROC BAD
390 GOTO 220
400 DEF PROC GOOD
410 *FX 15
420 CLS
430 COLOUR 12
440 PRINT"Congratulations"
450 PRINT"Happy landings!"
460 GOTO 530
470 DEF PROC BAD
480 CLS
490 COLOUR 11
500 *FX 15
510 PRINT"Splat! Squish! you"
520 PRINT"hit a building!"
530 PRINT:PRINT:PRINT
540 INPUT"Another go",G$
550 IF LEFT$(G$,1)="Y" THEN RUN
560 END
570 DEF PROC AIROPLANE
580 FOR X=1 TO 10
590 PRINTTAB(X-1,0);" "
600 PRINTTAB(X,0);CHR$(243);
610 SOUND 1,0,150+RND(50),3
620 SOUND 0,-15,3,3
630 T=TIME:REPEAT UNTIL TIME-T>10
640 NEXT X
650 FOR X=250 TO 220 STEP-5
660 SOUND 1,0,X,5

```

```
670      SOUND 0,-15,7,5
680      NEXTX
690      T=TIME:REPEAT UNTIL TIME-T>50
700      ENDPROC
```

>

PILOT

```
0 REM PILOT
10 MODE 7
20 PRINT TAB(10,10);CHR$(141);"Pilot"
30 PRINT TAB(10);CHR$(141);"Pilot"
40 A=INKEY(100)
50 MODE 4
60 VDU 19,0,4,0,0,0
70 VDU 23;8202;0;0;0;
80 PROC_DIALS
90 H=100+RND(300)
100 D1=30
110 DF=-1
120 S=80+RND(400)
130 TF=0
140 GCOL 0,7
150 PROC_INSTRUMENTS
160 GCOL 0,0
170 PROC_INSTRUMENTS
180 IF INKEY(-68)=-1 AND S<=500 THEN S
=S+RND(80)
190 IF INKEY(-82)=-1 AND S>0 THEN S=S-
RND(80)
200 IF INKEY(-51)=-1 AND H>0 THEN H=H-
RND(50):S=S+RND(5)
210 IF INKEY(-35)=-1 AND H<=500 THEN H
=H+RND(50):S=S-RND(5)
220 IF INKEY(-83)=-1 THEN DF=-DF
230 IF S<100 OR H<50 THEN SOUND 2,-15,
100,5
240 IF H>10 AND S<50+RND(5) THEN PRINT
"You stalled":GOTO 860
250 IF ABS(D1)>3 AND H<0 THEN PRINT"Yo
u missed the runway and crashed in a""p
otato field":GOTO 860
260 IF ABS(D1)<3 AND H<10 AND S<0 AND
TF=1 THEN PROC_GOOD
```

```

270 IF H>0 THEN TF=1
280 D1=D1+((S/1000)*DF)
290 SOUND 1,0,S/10+150,10
300 SOUND 0,-15,3,10
310 PROC_ALTER
320 IF H<0 THEN H=0
330 IF S<0 THEN S=0
340 GOTO 140
350 DEF PROC_DIALS
360 PROC_CIRCLE(200,800,200)
370 PRINT TAB(4,14); "SPEED"
380 PROC_CIRCLE(800,800,200)
390 PRINT TAB(20,14); "HEIGHT"
400 PROC_CIRCLE(200,300,200)
410 PRINTTAB(2,30); "DISTANCE"
420 PROC_CIRCLE(800,300,200)
430 PRINT TAB(20,30); "DIRECTION"
440 PROC_CIRCLE(1200,900,50)
450 PROC_CIRCLE(1200,600,50)
460 PROC_CIRCLE(1200,300,50)
470 PROC_CIRCLE(1200,80,50)
480 VDU 29,0;0;
490 MOVE 1279,1000
500 DRAW 1050,1000
510 DRAW 1050,0
520 DRAW 1279,0
530 PRINT TAB(35,6); "Tune"
540 PRINT TAB(34,15); "Volume"
550 PRINT TAB(35,25); "Tone"
560 PRINTTAB(0,30);
570 ENDPROC
580 DEF PROC_INSTRUMENTS
590 PRINT TAB(10,14); INT(S); " "; TAB(2
7,14); INT(H); " "; TAB(12,30); INT(D1); " "
600 SX=SIN(RAD(S)); SY=COS(RAD(S))
610 HX=SIN(RAD(H/2)); HY=COS(RAD(H/2))
620 DX=SIN(RAD(D1*10)); DY=COS(RAD(D1*1
0))
630 VDU 29,200;800;

```



```

640 MOVE 0,0
650 DRAW SX*150,SY*150
660 VDU 29,800;800;
670 MOVE 0,0
680 DRAW HX*150,HY*150
690 VDU 29,200;300;
700 MOVE 0,0
710 DRAW DX*150,DY*150
720 VDU 29,800;300;
730 MOVE -20,0
740 MOVE 20,0
750 IF DF=1 THEN PLOT 85,0,200
760 IF DF=-1 THEN PLOT 85,0,-200
770 ENDPROC
780 DEF PROC_CIRCLE (X,Y,R)
790 VDU 29,X;Y;
800 MOVE 0,R
810 FOR N=0 TO 360 STEP 20
820   X1=SIN(RAD(N)):Y1=COS(RAD(N))
830   DRAW X1*R,Y1*R
840   NEXTN
850 ENDPROC
860 FORD=1 TO 4000:NEXTD
870 MODE 4
880 *FX 15
890 VDU 29,0;0;
900 MOVE 500,500
910 DRAW 600,500
920 DRAW 600,700
930 DRAW 700,700
940 DRAW 700,800
950 DRAW 600,800
960 DRAW 600,900
970 DRAW 500,900
980 DRAW 500,800
990 DRAW 400,800
1000 DRAW 400,700
1010 DRAW 500,700
1020 DRAW 500,500

```

```

1030 PRINT TAB(15,8);"R.I.P."
1040 END
1050 DEF PROC_GOOD
1060 FOR D=1 TO 4000:NEXT D
1070 *FX 15
1080 SOUND 1,-15,0,5
1090 SOUND 1,-15,10,5
1100 SOUND 1,-15,0,2
1110 SOUND 1,-15,5,5
1120 SOUND 1,-15,20,15
1130 SOUND 1,-15,0,5
1140 CLS
1150 PRINTTAB(5,15);"Well done-a text b
ook landing!!!"
1160 END
1170 DEF PROC_ALTER
1180 A=RND(20)-10
1190 S=S-A
1200 A=RND(50)-25
1210 H=H-A
1220 ENDPROC
>

```

POEM

```

O REM POEM
10 MODE 7
20 PROCIN
30 VDU 14
40 R=RND(-TIME)
50 DATA run,walk,sleep,swim,play,fight,shine,weep,cook,fiddle with,snore,work,rest,paint,shoot,think,drive,kill,drink
60 DATA write,read,lead,plead,love,hate,hibernate,hear,smell,touch,see,define,fish,sneeze,recognise,put up a tent,finish,drown,fix,sell
70 DATA forget,remember,introduce,repair,seduce,prepare,resent,perspire,defend,defect,deduct,present,cry out,telephone,petrify
80 DATA receive,record,recall,cure,anger,take,make,bake,fake,forge,save,crave,steal,conquer,vandalise,explode,burn,destroy
90 DATA on,in,above,at,alongside,by,near,behind,far from,opposite,below,under,underneath,in front of,beside,against
100 DATA apple,grass,tree,flower,cow,bull,ant eater,dung,barn,stoat,hat,moat,stream,waterfall,water,sky,sun,moon,night,twilight
110 DATA hay,mayflower,straw,tree,wood,forest,sheep,slope,glade,cat,dog,heather,shepherd,farmer,house,heat,worm,soil,silence,air
120 DATA sea,lake,dew,branch,twig,fork,foot,web,spider,bee
130 DATA insect,ear,cricket,image,reflection,oxen,meat
140 DATA wool,sheep,ram
```

150 DATA leaf,lamb,snow,calf,smell,pig
,fence,field,corn,hill,mountain,barley,p
ath,church,village,real-ale

160 DATA cheese,malthouse,cider,sweat,
shear,plain,silence,river,bed,bread,rain
,star

170 DATA old,new,winding,bold,red,gree
n,cold,shining,frequent,fresh,light,bree
zy,heavy,ancient,warm,small,big,powerful
,wet

180 DATA slippery,cuddly,murderous,mer
ry,beautiful,ethnic,outspoken,purple,yel
low,pink,brown,green,white,black,blue,tu
rquoise

190 DATA gentle,mild,soft,quiet,serene
,dense,thick,desperate,destitute,bland,b
leak,modern,happy,gay,bright,dreary,bent
,warped

200 DATA invisible,clear,individual,ni
ce,weary,tired,sick,pristine,sparkling,w
ise,stupid,brainless,frank,strict,legles
s,utter

210 DATA Bang,Whoosh,Gosh,Holy-cow,Cr-
umbs,Crash,Crinkle,Drat,Clump

220 DIM V\$(72)

230 DIM P\$(16)

240 DIM N\$(87)

250 DIM A\$(70)

260 DIM E\$(9)

270 FOR V=1 TO 72

280 READ V\$(V)

290 NEXT

300 V=72

310 FOR P=1 TO 16

320 READ P\$(P)

330 NEXT

340 P=16

350 FOR N=1 TO 87

360 READ N\$(N)

```

370     NEXT
380 N=87
390 FOR A=1 TO 70
400     READ A$(A)
410     NEXT
420 A=70
430 FOR E=1 TO 9
440     READ E$(E)
450     NEXT
460 E=9
470 REPEAT
480     L=RND(8)
490     ON L GOSUB 520,570,650,700,760,7
80,850,900
500     IF RND(7)=7 THEN PRINT
510     UNTIL FALSE
520 PRINTN$(RND(N));"s ";
530 PRINTV$(RND(V));"ing ";
540 IF RND(10)>5 THEN PRINTP$(RND(P));
" ";
550 PRINTN$(RND(N));"s"
560 RETURN
570 PRINTP$(RND(P));" a ";
580 PRINTA$(RND(A));" ";
590 PRINTN$(RND(N));" ";
600 X=RND(10)
610 IF X>5 THEN PRINT"I ";ELSE PRINT"t
he ";N$(RND(N));" ";
620 PRINTV$(RND(V));
630 IF X>5 THEN PRINTELSE PRINT"s"
640 RETURN
650 PRINTE$(RND(E));"! ";
660 PRINT"the ";N$(RND(N));
670 IF RND(10)>5 THEN PRINT"s ";
680 PRINT" did ";V$(RND(V))
690 RETURN
700 REM
710 PRINTP$(RND(P));" ";
720 PRINTA$(RND(A));

```

```

730 IF RND(10)>7 THEN PRINT", ";;GOTO
720 ELSE PRINT" ";
740 PRINTN$(RND(N));"s "
750 RETURN
760 PRINTA$(RND(A));"ly ";
770 PRINTV$(RND(V));" the ";
780 PRINTA$(RND(A));" ";
790 PRINTN$(RND(N));"s"
800 RETURN
810 PRINTN$(RND(N));"s ";
820 PRINTV$(RND(V));"ing ";
830 PRINTA$(RND(A))
840 RETURN
850 PRINTA$(RND(A));" ";
860 PRINTV$(RND(V));"ing ";
870 PRINTP$(RND(P));" ";
880 PRINTN$(RND(N));"s"
890 RETURN
900 PRINTN$(RND(N));"s ";
910 PRINTV$(RND(V));" ";
920 PRINTP$(RND(P));" ";
930 PRINT"the ";N$(RND(N));
940 IF RND(10)>8 THEN PRINT"s" ELSE PR
INT
950 RETURN
960 DEF PROCIN
970 VDU 31,10,10,141
980 PRINT"Poem"
990 VDU 31,10,11,141
1000 PRINT"Poem"
1010 T=TIME
1020 REPEAT
1030 UNTIL INKEY(-99) OR TIME-T>200
1040 CLS
1050 ENDPROC
>

```

PORTALS

```
      0 REM PORTALS
      10 MODE6:VDU19,0,4,0,0,0
      20 MODE 7
      30 PROCIN
      40 MODE 4
      50 VDU 5
      60 PROCVA
      70 PROCH
      80 REPEAT
      90   PROCNB
     100   REPEAT
     110     PROCBG
     120     G%=G%+(INKEY(-83)-INKEY(-98))*
P
     130     D%=D%+P
     140     PROCBG
     150     P=D%/9
     160     UNTIL D%>500
     170     IF G%-E%/2>560 OR G%+E%/2<688 TH
EN PROCEX:END
     180     PROCBG
     190     UNTIL FALSE
     200 ENDPROC
     210 DEF PROCIN
     220 VDU 31,10,10,141
     230 PRINT"Portals"
     240 VDU 31,10,11,141
     250 PRINT"Portals"
     260 T=TIME
     270 REPEAT
     280   UNTIL INKEY(-99) OR TIME-T>200
     290 ENDPROC
     300 DEF PROCVA
     310 P=1
     320 VDU 23,255,255,255,255,255,255,255
,255,255
```

```

330 S%=0
340 ENDPROC
350 DEF PROCH
360 MOVE 560,528
370 VDU 255,255,255,255,255
380 ENDPROC
390 DEF PROCBD
400 ENDPROC
410 DEF PROCBG
420 GCOL 3,1
430 E%=D% DIV 2
440 MOVE G%-E%/2,512-E%
450 DRAW G%+E%/2,512-E%
460 DRAW G%+E%/2,512+E%
470 DRAW G%-E%/2,512+E%
480 DRAW G%-E%/2,512-E%
490 ENDPROC
500 DEF PROCNB
510 S%=S%+1
520 G%=640-RND(S%*30)+RND(S%*30)
530 IF G%>1200 THEN G%=1200
540 IF G%<0 THEN G%=0
550 VDU 19,1,RND(7),0,0,0
560 D%=10
570 PROCBG
580 P=S%
590 ENDPROC
600 DEF PROCEX
610 VDU 19,0,7,0,0,0
620 SOUND 0,-15,4,10
630 T=TIME
640 REPEAT UNTIL TIME-T>50
650 PROCH
660 VDU 20
670 FOR A%=1 TO 5
680     MOVE RND(1151),RND(1023)
690     PRINT"HA!"
700     VDU 19,1,RND(7),0,0,0
710     NEXT

```



```

720 VDU 4
730 S%=S%*5
740 PRINTTAB(0,0);"Score ";S%
750 IF H%<S% THEN H%=S%
760 PRINTTAB(0,2);"Hi-Score ";H%
770 PRINTTAB(0,10);"Play again? Y/N"
780 REPEAT
790   VDU 19,1,RND(7),0,0,0
800   SOUND RND(3),-3,RND(255),1
810   UNTIL INKEY(-69) OR INKEY(-86)
820 IF INKEY(-69) THEN RUN
830 CLS
840 *FX15,0
850 ENDPROC

```

>

RAT ATTACK

```
0 REM RAT ATTACK
10 MODE 7
20 VDU 31,10,10,141
30 PRINT"Rat Attack"
40 VDU 31,10,11,141
50 PRINT"Rat Attack"
60 T=TIME
70 REPEAT
80   UNTIL INKEY(-99) OR TIME-T>200
90 ON ERROR RUN
100 PX=7:MAN$=CHR$242:DIM RAT(4):FORT=
1104:RAT(T)=28-RND(10):NEXT:LV=10:SC=0
110 ENVELOPE1,3,-4,-1,2,6,6,28,81,-4,-
5,-1,126,63
120 MODE2:VDU23;8202;0;0;0;:*FX 11 0
130 PROCDEFINE
140 PROCGROUND
150 REPEAT
160   PROCMOVE
170   PROCDISPLAY
180   PROCRAT
190   PROCTOP
200   UNTIL LV=0
210 PRINTTAB(0,0)"THE WORLD HAS NOW  "
:PRINT"BEEN OVER RUN BY  ":PRINT"RATS,A
ND ITS ALL  ":PRINT"YOUR FAULT"
220 PRINT"YOU SCORED-";SC
230 *FX 15 0
240 PRINT:PRINT"PRESS ANY KEY"
250 A$=GET$
260 RUN
270 DEFPROCDEFINE
280 VDU23,240,16,56,84,84,124,124,124,
124,23,241,124,124,16,16,16,32,32,16
290 RAT$=CHR$240+CHR$10+CHR$8+CHR$241+
CHR$10+CHR$8+CHR$32
```

```

300 VDU23,242,224,227,231,72,112,64,64
,112,23,243,224,224,224,64,96,80,79,115,
23,244,199,199,39,18,14,2,2,14,23,245,7,
7,7,2,14,18,226,206,23,246,137,137,255,2
55,255,255,255,255
310 GRD#=CHR#246+CHR#246+CHR#246+CHR#3
2:GR#=STRING$(5,GRD#)
320 ENDPROC
330 DEFPROCGROUND:COLOUR2
340 VDU19,0,4;0;
350 FORT=31 TO 10 STEP-1
360 PRINTTAB(0,T);GR#
370 NEXT
380 DEFPROCMOVE
390 A#=INKEY$(0):IFA#=""THEN ENDPROC
400 IFA#="Z" AND PX>1 THEN MAN#=CHR#24
4:PX=PX-1
410 IFA#="C" ANDPX<17 THEN MAN#=CHR#24
2:PX=PX+1
420 IFA#="/" AND MAN#<>CHR#245 AND MAN
#<>CHR#243 THEN MAN#=CHR$(ASC(MAN#)+1):P
ROCSCAN:ENDPROC
430 IFA#="/" AND MAN#<>CHR#244 AND MAN
#<>CHR#242 THEN MAN#=CHR$(ASC(MAN#)-1):P
ROCSCAN
440 ENDPROC
450 DEFPROCDISPLAY:COLOUR3
460 PRINTTAB(PX,9);" ";MAN#;" "
470 ENDPROC
480 DEFPROCRAI:COLOUR1
490 FORT=1 TO 4
500 IF RND(10)<2THEN RAT(T)=RAT(T)-1
:IF RAT(T) <9 THEN PROCOUT
510 NEXTT
520 PRINTTAB(3,RAT(1));RAT#
530 PRINTTAB(7,RAT(2));RAT#
540 PRINTTAB(11,RAT(3));RAT#
550 PRINTTAB(15,RAT(4));RAT#
560 SOUND1,-6,10,1

```

```

570 COLOUR2:ENDPROC
580 DEFPROCOUT
590 SOUND0,-15,50,1
600 PRINTTAB(3*T+(T-1),10);" ";TAB(3*T
+ (T-1),9);" "
610 RAT(T)=28-RND(15):LV=LV-1
620 ENDPROC
630 DEFPROCSCAN
640 IF PX=2 AND RAT(1) <=10 THEN R=1:P
ROCSPLAT
650 IF PX=6 AND RAT(2) <=10 THEN R=2:P
ROCSPLAT
660 IF PX=10 AND RAT(3) <=10 THEN R=3:
PROCSPLAT
670 IF PX=14 AND RAT(4) <=10 THEN R=4:
PROCSPLAT
680 ENDPROC
690 DEFPROCSPAT
700 PRINTTAB(PX+1,10);" ";TAB(PX+1,11)
;" "
710 SOUND1,1,100,10
720 SC=SC+RND(5)*10:RAT(R)=28-RND(15)
730 ENDPROC
740 DEFPROCTOP:COLOUR7
750 PRINTTAB(0,0)"SCORE-";SC;"    LIVES
-";LV;" "
760 ENDPROC
>

```

SHARK

```
0 REM SHARK
10 MODE7
20 PRINTTAB(10,10);CHR$(141);"Shark"
30 PRINTTAB(10);CHR$(141);"Shark"
40 A=INKEY(100)
50 MODE 2
60 VDU 23,244,28,28,8,127,8,20,34,65
70 VDU 23,245,0,16,16,191,255,56,0,0
80 VDU 23,246,255,255,255,255,255,255,255,255
,255,255
90 VDU 23,247,0,8,28,28,255,254,252,2
48
100 VDU 23;8202;0;0;0;0;
110 COLOUR 3
120 PRINTTAB(10,2);CHR$(247)
130 FOR A=3 TO 30
140   FOR B=0 TO 19
150     IF B<2 THEN COLOUR 2 ELSE COLO
UR 4
160     PRINTTAB(B,A);CHR$(246);
170     NEXTB,A
180     PX=RND(15)+2
190     PY=3+RND(27)
200     SX=RND(15)+2
210     SY=3+RND(27)
220     COLOUR 132
230     COLOUR 1
240     PRINTTAB(PX,PY);CHR$(244)
250     COLOUR 7
260     PRINTTAB(SX,SY);CHR$(245)
270     T=TIME:REPEAT UNTIL TIME-T>100
280     COLOUR 4
290     PRINTTAB(PX,PY);CHR$(246)
300     PRINTTAB(SX,SY);CHR$(246)
```

```

310     SX=PX+(RND(8)-4)
320     SY=PY+(RND(8)-4)
330     SOUND 1,-15,ABS(SX-PX)*20,5
340     SOUND 2,-15,ABS(SY-PY)*20,5
350     IF SX<4 THEN SX=4
360     IF SX>19 THEN SX=19
370     IF SY<3 THEN SY=3
380     IF SY>30 THEN SY=30
390     IF INKEY(-98)=-1 THEN PX=PX-1
400     IF INKEY(-83)=-1 THEN PX=PX+1
410     IF INKEY(-105)=-1 THEN PY=PY+1
420     IF INKEY(-73)=-1 THEN PY=PY-1
430     IF PX>19 THEN PX=19
440     IF PY<4 THEN PY=4
450     IF PY>30 THEN PY=30
460     IF PX=SX AND PY=SY THEN PROCEAT
470     IF PX<3 THEN PROCLAND
480     GOTO 230
490     DEF PROCLAND
500     COLOUR 7
510     CLS
520     SOUND 1,-15,100,5
530     PRINT"I'll get you next"
540     PRINT"time!!!"
550     VDU30
560     T=TIME:REPEAT UNTIL TIME-T>100
570     PRINT"*****"
580     PRINT"*****"
590     T=TIME:REPEAT UNTIL TIME-T>100
600     VDU 30
610     GOTO 520
620     DEF PROCEAT
630     COLOUR 7
640     CLS
650     SOUND 1,-15,0,5
660     PRINT"You taste very nice!"
670     VDU 30

```

```
680  T=TIME:REPEAT UNTIL TIME-T>100
690  PRINT"*****"
700  T=TIME:REPEAT UNTIL TIME-T>100
710  VDU 30
720  GOTO 650
```

>

SHOCK

```
0 REM SHOCK
10 MODE 7
20 PROCIN
30 MODE 2
40 PROCVA
50 PROCB0
60 PROCUS
70 PROCPS
80 PROCRC
90 PROCPL
100 REPEAT
110   PROCMS
120   IF NOT D% THEN PROCSD
130   IF D% THEN PROCMD
140   IF Q%>-1 THEN PROCMS:IF F% THEN
PROCMB
150   IF INKEY(-74) THEN PROCSE
160   IF F% THEN PROCMB
170   IF Q%>-1 THEN PROCMR
180   IF NOT D1% AND RND(100)=1 THEN P
ROCSN
190   IF D1% THEN PROCMN
200   UNTIL FALSE
210 END
220 DEF PROCIN
230 VDU 31,10,10,141
240 PRINT"Shock"
250 VDU 31,10,11,141
260 PRINT"Shock"
270 T=TIME
280 REPEAT
290   UNTIL INKEY(-99) OR TIME-T>200
300 ENDPROC
310 DEF PROCVA
320 VDU 5
330 U%=8
```



```

340 P%=40
350 DIM I(10)
360 B%=0
370 F%=0
380 D%=0
390 Q%=-1
400 DIM M(9)
410 DIM N(9)
420 DIM C(9)
430 DIM R(10)
440 L$=CHR$255+CHR$255+CHR$255
450 S%=0
460 N$=""
470 D1%=0
480 C1%=0
490 ENDPROC
500 DEF PROCBD
510 GCOL 0,3
520 FOR A%=159 TO 1119 STEP 80
530     IF A%=639 THEN 560
540     MOVE A%,767
550     DRAW A%,511
560     NEXT
570 MOVE 0,0
580 DRAW 1279,0
590 DRAW 1279,1023
600 DRAW 0,1023
610 DRAW 0,0
620 ENDPROC
630 DEF PROCUS
640 VDU 23,255,0,0,24,24,60,126,255,25
5
650 VDU 23,254,24,126,255,153,153,255,
60,24
660 VDU 23,253,24,0,60,0,126,0,255,0
670 VDU 23,252,10,31,63,85,85,170,255,
73
680 VDU 23,251,80,248,252,170,170,85,2
55,146

```

```

690 VDU 23,250,66,36,102,24,60,129,66,
60
700 ENVELOPE 1,1,50,25,10,12,12,12,100
,1,0,-2,100,100
710 ENVELOPE 2,3,-3,5,-2,6,6,6,25,-5,0
,-8,100,75
720 ENDPROC
730 DEF PROCPS
740 GCOL 3,1
750 MOVE U%,P%
760 VDU 255
770 ENDPROC
780 DEF PROCSEB
790 IF F% THEN ENDPROC
800 F%=-1
810 L%=U%+32
820 E%=P%+32
830 PROCPB
840 ENDPROC
850 DEF PROCPB
860 GCOL 3,7
870 MOVE L%,E%
880 DRAW L%,E%+32
890 ENDPROC
900 DEF PROCMB
910 PROCPB
920 E%=E%+64
930 IF POINT(L%,E%)<>0 THEN PROCHT:END
PROC
940 PROCPB
950 ENDPROC
960 DEF PROCMS
970 PROCPS
980 U%=U%+(INKEY(-98)-INKEY(-83))*16
990 PROCPS
1000 ENDPROC
1010 DEF PROCSD
1020 D%=-1
1030 C%=RND(10)-1

```

```

1040 PROCO(C%)
1050 I%=Y%
1060 V%=1023
1070 PROCPD
1080 ENDPROC
1090 DEF PROCPD
1100 GCOL 3,6
1110 MOVE I%,V%
1120 VDU 254
1130 ENDPROC
1140 DEF PROCMD
1150 PROCPD
1160 V%=V%-32+(V%>703)*32
1170 PROCPD
1180 IF V%=543+32*I(C%) THEN PROCED:END
PROC
1190 ENDPROC
1200 DEF PROCO(X%)
1210 IF X%<5 THEN Y%=167+X%*80 ELSE Y%=
727+(X%-5)*80
1220 ENDPROC
1230 DEF PROCED
1240 D%=0
1250 I(C%)=I(C%)+1
1260 IF I(C%)<>5 THEN SOUND 2,-10,I(C%)
*20,5:ENDPROC
1270 Q%=Q%+1
1280 J%=C%
1290 PROCSR
1300 PROCO(C%)
1310 M(Q%)=Y%
1320 N(Q%)=479
1330 R(Q%)=C%
1340 W%=Q%
1350 PROCPR
1360 ENDPROC
1370 DEF PROCPC
1380 PROCRC
1390 S%=S%+50

```

```

1400 PROCRC
1410 ENDPROC
1420 DEF PROCRC
1430 GCOL 3,3
1440 MOVE 8,900
1450 PRINT ;S%
1460 ENDPROC
1470 DEF PROCPL
1480 MOVE 0,1000
1490 GCOL 3,1
1500 PRINT L$
1510 ENDPROC
1520 DEF PROCHT
1530 G%=POINT(L%,E%)
1540 F%=0
1550 IF D1% THEN IF ABS(L%-A1%-32)<32 T
HEN D1%=0:C1%=0:SOUND 16,0,0,1:PROCPN
1560 IF G%<>>6 THEN SOUND D1%,1,-10,1:EN
DPROC
1570 PROCPN
1580 IF L%<640 THEN J%=(L%-167)/80 ELSE
J%=(L%-727)/80+5
1590 IF I(J%)=0 THEN SOUND 2,-17,200,5:
ENDPROC
1600 DEF PROCSR
1610 GCOL 0,0
1620 PROCO(J%)
1630 MOVE Y%,511+32*I(J%)
1640 VDU 254
1650 I(J%)=I(J%)-1
1660 SOUND 1,1,4,1
1670 ENDPROC
1680 DEF PROCMR
1690 W%=-1
1700 REPEAT
1710   W%=W%+1
1720   PROCPR
1730   N(W%)=N(W%)-32
1740   IF N(W%)=31 THEN PROCER:UNTIL W%
>=Q%:ENDPROC

```

```

1750 PROCPR
1760 UNTIL W%>=Q%
1770 ENDPROC
1780 DEF PROCPR
1790 GCOL 3,6
1800 MOVE M(W%),N(W%)
1810 VDU 254
1820 ENDPROC
1830 DEF PROCER
1840 GCOL 3,5
1850 MOVE M(W%)-64,N(W%)
1860 VDU 253,254,253
1870 IF NOT D1% THEN SOUND 0,-15,3,10
1880 Q%=Q%-1
1890 MOVE M(W%)-64,N(W%)
1900 VDU 253,254,253
1910 HI%=ABS(U%-M(W%))<128
1920 IF Q%=-1 THEN 1980
1930 FOR Z%=0 TO Q%
1940   R(Z%)=R(Z%+1)
1950   M(Z%)=M(Z%+1)
1960   N(Z%)=N(Z%+1)
1970 NEXT Z%
1980 IF NOT Q% THEN W%=W%-1
1990 IF HI%=FALSE THEN ENDPROC
2000 PROCPL
2010 L#=LEFT$(L$,LENL$-1)
2020 PROCPL
2030 SOUND 3,-15,200,20
2040 VDU 19,0,7,0,0,0
2050 VDU 19,0,0,0,0,0
2060 IF LENL$>0 THEN ENDPROC
2070 *FX15,0
2080 VDU 4
2090 IF H%<S% THEN H%=S%
2100 VDU 31,6,5
2110 PRINT"Game Over"
2120 VDU 31,6,20
2130 PRINT"Score ";S%

```

```

2140 VDU 31,4,25
2150 PRINT"Hi-Score ";H%
2160 COLOUR 1
2170 VDU 31,2,30
2180 PRINT"Play again? Y/N";
2190 REPEAT
2200   UNTIL INKEY(-69) OR INKEY(-86)
2210 IF INKEY(-69) THEN RUN
2220 *FX15,0
2230 CLS
2240 END
2250 DEF PROCMN
2260 C1%=C1%+1
2270 IF C1%<25 THEN ENDPROC
2280 C1%=0
2290 GCOL 0,10
2300 MOVE 8,846
2310 N$=N$+CHR$254
2320 PRINTN$
2330 IF LENN$=20 THEN PROCLN:ENDPROC
2340 ENDPROC
2350 DEF PROCSN
2360 D1%=-1
2370 SOUND 0,2,4,-1
2380 A1%=(RND(3)-1)*600+8
2390 DEF PROCPN
2400 MOVE A1%,814
2410 GCOL 3,13
2420 VDU 250
2430 ENDPROC
2440 DEF PROCLN
2450 N$=""
2460 GCOL 0,0
2470 MOVE 8,846
2480 PRINT STRING$(20,CHR$254)
2490 D1%=0
2500 PROCPN
2510 SOUND 16,0,1,1
2520 GOTO 2000

```

SLAVES

```
0 REM SLAVES
10 MODE 7
20 PROCIN
30 PROCVA
40 FOR Y%=1 TO 20
50   PROCY
60   A$="sell"
70   PROCBY
80   A$="buy"
90   PROCBY
100  NEXT
110 PROCEN
120 PROCAG
130 END
140 DEF PROCIN
150 VDU 31,10,10,141
160 PRINT"Slaves"
170 VDU 31,10,11,141
180 PRINT"Slaves"
190 VDU 31,0,23,129
200 PRINT"Please ensure that caps lock
is OFF"
210 T=TIME
220 REPEAT
230   UNTIL INKEY(-99) OR TIME-T>100
240 ENDPROC
250 DEF PROCVA
260 DIM R(30)
270 DIM C$(30)
280 DIM P(30)
290 FOR A%=1 TO 30
300   READ C$(A%)
310   IF RND(10)>7 THEN P(A%)=RND(1000
0)
320   NEXT
330 P(0)=RND(10000)+10000
```

```

340 FOR A%=1 TO 30
350   READ R(A%)
360   NEXT
370 PROCPF(0)
380 ENDPROC
390 DEF PROCP
400 PRINT "You have: " '
410 PROCPR(P(0))
420 PRINT ;R$'
430 FOR A%=1 TO 30
440   IF P(A%)>0 THEN PRINT ;P(A%),C$(
A%)
450   NEXT
460 ENDPROC
470 DEF PROCY
480 PROCDS
490 PRINT "LIGHT YEARS ";Y%
500 PRINTSTRING$(40,"*")
510 PRINT
520 PROCP
530 ENDPROC
540 DEF PROCBY
550 PRINT
560 B%=RND(5)-1
570 IF B%<>1 THEN PRINT "There are ";B
%; " traders" ELSE PRINT "There's just 1
trader"
580 PRINT "wishing to ";A$;" some goods
."
590 PRINT
600 VDU 31,0,23,132,157,135,136,9,9,9,
9
610 PRINT "Please hit the space-bar";
620 IF INKEY(-99) THEN REPEAT:UNTIL NO
T INKEY(-99)
630 REPEAT
640   UNTIL INKEY(-99)
650 IF B%=0 THEN CLS:ENDPROC
660 FOR C%=1 TO B%

```



```

670    *FX15,0
680    PROCDS
690    G%=RND(30)
700    IF RND(2)-2 THEN PROCCH
710    IF G%=0 THEN NEXT:CLS:ENDPROC
720 PRINT"Trader ";C%;" wants to ";A$
730 M%=RND(5000)
740 X%=RND(10000)+M%
750 PRINT ;M%;" to ";X%
760 PRINT;C$(G%)
770 R=R(G%)*RND(100)/100
780 PROCPR(R)
790 PRINT"at ";R%;" each"
800 PRINT'"How many to ";
810 IF A$="sell" THEN PRINT"buy?" ELSE
PRINT"sell?"
820 PRINT"(You have ";
830 IF A$="sell" THEN PROCIS ELSE PROC
IB
840 NEXT
850 ENDPROC
860 DEF PROCDS
870 VDU 30
880 FOR E%=0 TO 24
890    PROCWT(2)
900    VDU 11
910    NEXT
920 ENDPROC
930 DEF PROCIS
940 PROCPR(P(0))
950 PRINT ;R%;" )"
960 INPUT U%
970 IF U%>X% THEN PROCX:GOTO 960
980 IF U%<M% AND U%<>0 THEN PROCM:GOTO
960
990 IF A$="buy" THEN U%=-U%
1000 IF P(0)-U%*R<0 THEN PROCNM:GOTO 96
0
1010 IF P(G%)+U%<0 THEN PROCNG:GOTO 960

```

```

1020 P(0)=P(0)-U%*R
1030 P(G%)=P(G%)+U%
1040 ENDPROC
1050 DEF PROCIB
1060 PRINT ;P(G%)
1070 PRINT ;C$(G%);")"
1080 GOTO 960
1090 DEF PROCWT(K)
1100 T=TIME
1110 REPEAT
1120   UNTIL TIME-T>K
1130 ENDPROC
1140 DEF PROCM
1150 VDU 7,10
1160 PRINT"The minimum ";C$(G%)
1170 PRINT"this trader can ";A$;" is ";
M%'
1180 ENDPROC
1190 DEF PROCX
1200 VDU 7,10
1210 PRINT"The most ";C$(G%)
1220 PRINT"the geezer is willing to ";A
$;" is ";X%'
1230 ENDPROC
1240 DEF PROCNM
1250 VDU 7,10
1260 PROCPR(U%*R)
1270 PRINT "You ain't got enough money,
kiddo"
1280 PRINT;U%;" ";C$(G%)
1290 PRINT"will cost you ";R$
1300 PROCPR(P(0))
1310 PRINT"and you only have ";R$
1320 ENDPROC
1330 DEF PROCNG
1340 VDU 7,10
1350 PRINT "You can't sell more goods"
1360 PRINT "you've got!"'
1370 ENDPROC

```

```

1380 DEF PROCPR(Y)
1390 R$=""
1400 Q=INT(Y*100)/100
1410 IF Q>=1 THEN R$=" "+STR$(Q)
1420 IF Q<1 THEN R$=STR$(Q*100)+"p"
1430 IF Q=0 THEN R$="\p":R=0.005
1440 ENDPROC
1450 DEF PROC PF(T%)
1460 V=P(O)
1470 FOR A%=1 TO 30
1480     V=V+P(A%)*R(A%)*2
1490     NEXT
1500 IF T%=0 THEN I=V ELSE J=V
1510 ENDPROC
1520 DEF PROCEN
1530 FOR A%=1 TO 12
1540     PRINT"YOU HAVE ARRIVED IN PORT"
1550     NEXT
1560 PROCWT(50)
1570 PROC PF(-1)
1580 PROCDS
1590 PROCPR(I)
1600 PRINT"You started out with ";R$
1610 PRINT"worth of goods and money"
1620 PRINT'
1630 PROCPR(J)
1640 PRINT"You now have ";R$;" worth."
1650 PRINT
1660 F%=(J-I)/I*100
1670 PRINT"Profit = ";F%;"%"
1680 PRINT'
1690 IF F%>H% THEN H%=F%
1700 PRINT"Best so far is ";H%
1710 ENDPROC
1720 DEF PROCAG
1730 PRINT'''
1740 PRINT"Another go? (press 'Y' or 'N'
')"
1750 REPEAT

```

```

1760   UNTIL INKEY(-69) OR INKEY(-86)
1770 *FX15,1
1780 IF INKEY(-69) THEN RUN
1790 CLS
1800 ENDPROC
1810 DEF PROCCH
1820 PRINT" 'What do you want to ";
1830 IF A$="buy" THEN PRINT"sell?" ELS
E PRINT"buy?"
1840 PRINT"says Trader ";C%
1850 INPUT Y$
1860 IF Y$="" THEN G%=0:ENDPROC
1870 J%=0
1880 REPEAT
1890   J%=J%+1
1900   IF LENY$>LENC$(J%) THEN 1890:REM
THIS LINE IS VERY IMPORTANT IF
YOU HAVE A 1.0 ISSUE BBC (SEE PA
GE
281 OF USER GUIDE)
1910   UNTIL INSTR(C$(J%),Y$)<>0 OR J%=
30
1920 IF J%=30 THEN PRINT" 'What's ";Y$;
"?":GOTO 1820
1930 PRINT" 'Did you say ";C$(J%);"?
Y/N"
1940 REPEAT
1950   UNTIL INKEY(-69) OR INKEY(-86)
1960 IF INKEY(-86) THEN 1880
1970 *FX15,0
1980 G%=J%
1990 PROCDS
2000 ENDPROC
2010 DATA pieces of orange peel, tonnes
of coal dust, kilograms of grain, ex-dicta
tors, test-tubes, paper bags, pen tops, slip
pers, missiles, C-60 cassettes, ZX80's, spon
ge bags, spare ribs, smarties, plastic bric
ks, gallons of silicon ink

```

2020 DATA dominoes,oily sprockets,wobbly stools,tins of gloss paint,unfloppy disks,search warrants,badges,paper aeroplanes,dead trolleys,bars of reject soap,used 13amp fuses,postcards from Milton-Keynes,flower arrangers,souvenirs

2030 DATA .7,12.83,2.23,23.02,1.97,.11,.21,7.34,976.41,5.41,17.5,3.12,.99,.78,.17,57.92,.75,8.46,4.01,9.99,75.46,215.54,.32,3.67,21,5.09,.27,3.38,75.19,6.64

>

SMASH CHAMBER

```
0 REM SMASH CHAMBER
10 MODE 6:VDU 19,0,4,0,0,0
20 MODE 7
30 VDU 31,10,11,141
40 PROCIN
50 PROCVA
60 MODE 2
70 VDU 5
80 *FX9,1
90 *FX10,1
100 REPEAT
110   IF L%>1 THEN GOSUB 380
120   PROCBO
130   GOSUB 990
140   REPEAT
150     SOUND 1,-10,V%,1
160     PROCPS
170     PROCPC
180     PROCPS
190     PROCDI
200     PROCD
210     UNTIL POINT(U%,F%)<>0
220   PROCNS
230   UNTIL L%=5
240 GOSUB 1260
250 *FX15,0
260 END
270 DEF PROCIN
280 VDU 31,10,10,141
290 PRINT"Smash Chamber"
300 VDU 31,10,11,141
310 PRINT"Smash Chamber"
320 T=TIME
330 REPEAT
340   UNTIL INKEY(-99) OR TIME-T>200
350 ENDPROC
```

```

360 DEF PROCVA
370 L%=1
380 U%=640
390 P%=512
400 D%=3
410 G%=-1
420 V%=16
430 IF L%>1 THEN RETURN
440 S%=0
450 ENVELOPE 1,1,3,-3,3,100,200,100,10
0,0,0,-100,100,100
460 VDU 23,255,24,24,24,60,199,255,60,
24
470 VDU 23,254,128,192,96,248,96,254,1
27,254
480 VDU 23,253,24,60,255,199,60,24,24,
24
490 VDU 23,252,1,3,6,31,6,127,254,127
500 ENDPROC
510 DEF PROCBO
520 FOR A%=1 TO 15+L%*5
530   GCOL 0,RND(7)
540   X%=RND(1280)
550   Y%=RND(1024)
560   F%=RND(32)+16
570   MOVE X%,Y%
580   MOVE X%+F%,Y%
590   PLOT 85,X%,Y%+F%
600   PLOT 85,X%+F%,Y%+F%
610   NEXT
620 GCOL 0,4
630 MOVE 0,1024
640 MOVE 1280,1024
650 PLOT 85,0,960
660 PLOT 85,1280,960
670 MOVE 0,1008
680 GCOL 0,7
690 PRINT "Level ";L%
700 MOVE 640,1008

```

```

710 PRINT "Score"
720 PROC SH
730 ENDPROC
740 DEF PROC SH
750 FOR A%=0 TO 512 STEP 12
760     GCOL 3,RND(7)
770     MOVE A%+64,1024-A%
780     FOR B%=1 TO 2
790         DRAW 1216-A%,1024-A%
800         DRAW 1216-A%,A%
810         DRAW A%+64,A%
820         DRAW A%+64,1024-A%
830     NEXT
840     SOUND 1,-10,A%/3,1
850 NEXT
860 ENDPROC
870 DEF PROC PS
880 MOVE U%,P%
890 GCOL 3,8+L%
900 PRINTCHR$(256-D%)
910 ENDPROC
920 DEF PROC D
930 T%=D%
940 D%=-INKEY(-73)-(INKEY(-83))*2-(INKEY(-105))*3-(INKEY(-98))*4
950 IF ABS(D%-T%)=2 AND D%<>0 THEN PROC CPE
960 IF D%=0 OR D%>4 THEN D%=T%
970 ENDPROC
980 DEF PROC PC
990 MOVE 1024,1008
1000 GCOL 3,3
1010 PRINT ;S%
1020 IF G%=-1 THEN G%=0:RETURN
1030 S%=S%+1
1040 IF S%/10=S% DIV 10 THEN V%=V%+1
1050 MOVE 1024,1008
1060 PRINT ;S%
1070 ENDPROC

```



```

1080 DEF PROCNS
1090 PROCEX
1100 CLG
1110 V%=16
1120 L%=L%+1
1130 ENDPROC
1140 DEF PROCPE
1150 SOUND 1,1,100,10
1160 V%=V%+4
1170 ENDPROC
1180 DEF PROCEX
1190 SOUND 0,-L%*3,5,50
1200 FOR A%=1 TO 120
1210   GCOL 0,RND(3)
1220   MOVE U%,P%
1230   DRAW RND(1280),RND(1024)
1240   NEXT
1250 ENDPROC
1260 MODE 7
1270 IF H%<S% THEN H%=S%
1280 VDU 31,16,11,141
1290 PRINT"Game Over"
1300 VDU 31,16,12,141
1310 PRINT"Game Over"
1320 VDU 31,0,20,132,157,135
1330 PRINT"Score ";S%,"... 'nother go?
(Yes or No)"
1340 VDU 31,0,22,129,136
1350 PRINT"Hi-Score ";H%
1360 REPEAT
1370   UNTIL INKEY(-69) OR INKEY(-86)
1380 IF INKEY(-69) THEN RUN
1390 CLS
1400 RETURN
1410 DEF PROCDI
1420 IF D%>4 THEN D%=RND(4)
1430 RESTORE 1510
1440 FOR A%=1 TO D%
1450   READ R%

```

```
1460 READ E%  
1470 NEXT  
1480 U%=U%+R%*V%  
1490 P%=P%+E%*V%  
1500 ENDPROC  
1510 DATA 0,1,1,0,0,-1,-1,0  
>
```

SYNTHESIZER

```
0 REM SYNTHESIZER
10 MODE 7
20 PRINT "Which sound effect do you want?"
30 PRINT "1.....Piano"
40 PRINT "2.....Space Invaders"
50 INPUT S1
60 IF S1<1 OR S1>2 THEN 50
70 INPUT "Do you wish to 'enhance' the sound", I1$
80 PRINT TAB(10);CHR$(134);CHR$(141);"Synthesizer."
90 PRINT TAB(10);CHR$(134);CHR$(141);"Synthesizer."
100 PRINT TAB(10);CHR$(131);CHR$(141);"-----"
110 PRINT TAB(10);CHR$(131);CHR$(141);"-----"
120 PRINT
130 IF S1=2 THEN ENVELOPE 1,1,-128,127,0,1,1,1,126,-10,0,-10,126,126 ELSE ENVELOPE 1,1,0,0,0,1,1,1,126,-5,0,-5,126,126
140 A=0
150 B=0
160 C=FNGET
170 C=C*5
180 IF LEFT$(I1$,1)="Y" THEN SOUND 0,1,A,5
190 SOUND 1,1,A,5
200 SOUND 2,1,B,5
210 SOUND 3,1,C,5
220 B=A:A=C
230 GOTO 160
240 DEF FNGET
250 RESTORE
```

```

260 D$=GET$:PRINTCHR$(128+RND(7));CHR$(136);D$;:IF D$<"0"THEN260
270 IF D$>"9" AND D$<"A" THEN 260
280 IF D$>"Z" THEN 260
290 REPEAT READ A$,N
300 UNTIL D$=A$:=N
310 NEXTA
320 DATA 1,1,2,2,3,3,4,4,5,5,6,6,7,7,8
,8,9,9,0,0,0,10,W,11,E,12,R,13,T,14,Y,15
,U,16,I,17,0,18,P,19,A,20,S,21,D,22,F,23
,G,24,H,25,J,26,K,27,L,28,Z,29,X,30,C,31
,V,32,B,33,N,34,M,35
>

```

TARGET

```
0 REM TARGET
10 MODE 7
20 PRINTTAB(10,10);CHR$(141);"Target"
30 PRINTTAB(10);CHR$(141);"Target"
40 A=INKEY(100)
50 *TV 255
60 MODE 5
70 VDU 23;8202;0;0;0;
80 VDU 23,244,24,24,24,24,255,255,255
,255
90 VDU 23,245,255,189,189,189,24,36,6
6,129
100 VDU 23,246,24,24,24,24,24,24,24,24
110 VDU 23,247,255,189,189,189,24,36,3
6,36
120 PRINT"Speed"
130 PRINT"(1=Blitz,20=Slow)"
140 INPUTTAB(5,0),V%
150 CLS
160 VDU 19,2,2,0,0,0,19,3,3,0,0,0
170 PROC_STAR
180 L%=0
190 S%=0
200 H%=0
210 X%=RND(20)-1
220 P%=RND(20)-1
230 D%=RND(2)
240 IF D%=2 THEN D%=-1
250 COLOUR 3
260 IF L%=0 THEN PRINT TAB(F%,0);CHR$(
245):L%=1:GOTO290
270 PRINT TAB(P%,0);CHR$(247)
280 L%=0
290 COLOUR 2
300 PRINTTAB(X%,20);CHR$(244)
310 IF INKEY(-74)=-1 THEN PROC_FIRE
```

```

320 COLOUR 1
330 PRINT TAB(0,25);"Fired:";S%
340 PRINT TAB(0,30);"Hit:";H%
350 PLOT69,RND(1279),RND(1023)
360 T%=TIME:REPEAT UNTIL TIME-T%>V%
370 COLOUR 0
380 PRINTTAB(P%,0);" "
390 PRINTTAB(X%,20);" "
400 X%=X%+INKEY(-98)-INKEY(-83)
410 IF X%<0 THEN X%=0
420 IF X%>19 THEN X%=19
430 P%=P%+D%
440 SOUND 1,-15,P%*10,1
450 IF S%>30 THEN PROC_END
460 IF P%>19 OR P%<0 THEN 220
470 IF RND(10)=1 THEN 230 ELSE GOTO 25
0
480 DEF PROC_FIRE
490 B%=X%
500 I%=B%
510 FOR Y%=18 TO 1 STEP -1
520     SOUND 2,-15,200-(Y%*10),1
530     PRINTTAB(B%,Y%+1);" "
540     PRINTTAB(X%,Y%);CHR$(246)
550     A%=INKEY(2)
560     B%=X%
570     X%=X%+INKEY(-98)-INKEY(-83)
580     IF X%>19 THEN X%=19
590     IF X%<0 THEN X%=0
600     NEXT Y%
610 PRINTTAB(B%,1);" "
620 IF X%=P% THEN PROC_HIT ELSE PROC_M
ISS
630 X%=I%
640 ENDPROC
650 DEF PROC_HIT
660 *FX 15,1
670 S%=S%+1
680 H%=H%+1

```

```

690 FOR A%=250 TO 220 STEP -1
700   SOUND 1,0,A%,1
710   SOUND 0,-15,7,1
720   NEXTA%
730 FOR A%=1 TO 20
740   PRINTTAB(P%,0); "*"
750   K%=INKEY(10)
760   PRINTTAB(P%,0); CHR$(245)
770   K%=INKEY(10)
780   NEXTA%
790 ENDPROC
800 DEF PROC_MISS
810 S%=S%+1
820 SOUND 1,0,200,5
830 SOUND 0,-15,3,5
840 ENDPROC
850 DEF PROC_END
860 CLS
870 COLOUR 3
880 PRINT "You hit "; H%; " UFO's"
890 *FX 15
900 PRINT "Another go?";
910 A$=GET$
920 PRINTTAB(11,1); A$
930 IF A$<>"Y" AND A$<>"N" THEN 910
940 IF A$="Y" THEN RUN
950 END
960 DEF PROC_STAR
970 FOR A%=1 TO 100
980   PLOT 69,RND(1279),RND(1023)
990   NEXTA%
1000 ENDPROC

```

>

UPS 'N' DOWNS

```
0 REM UPS 'N' DOWNS
10 MODE 7:VDU 23;8202;0;0;0;0;
20 VDU 23,244,0,15,15,15,15,15,15,15
30 VDU 23,245,0,240,240,240,240,240,2
40,240
40 PRINTTAB(10,10);CHR$(141);CHR$(129
);"Ups 'n' Downs"
50 PRINTTAB(10);CHR$(141);CHR$(129);"
Ups 'n' Downs"
60 A=INKEY(100):MODE 2
70 COLOUR 1:PRINT" Ups 'n' Downs":C
OLOUR7
80 GCOL 0,3
90 MOVE 0,0:DRAW 1279,0:DRAW 1279,100
:DRAW 0,100:DRAW 0,0
100 FORX=0 TO 1279 STEP 128
110 MOVEX,0:DRAW X,100:NEXTX
120 MOVE 0,200:DRAW 1279,200:DRAW 1279
,300:DRAW 0,300:DRAW 0,200
130 FORX=0 TO 1279 STEP 128
140 MOVEX,200:DRAW X,300:NEXTX
150 MOVE 0,400:DRAW 1279,400:DRAW 1279
,500:DRAW 0,500:DRAW 0,400
160 FORX=0TO1279 STEP 128:MOVEX,400:DR
AW X,500:NEXTX
170 MOVE 0,600:DRAW 1279,600:DRAW 1279
,700:DRAW 0,700:DRAW 0,600
180 FORX=0TO1279 STEP 128:MOVEX,600:DR
AW X,700:NEXTX
190 MOVE 0,800:DRAW 1279,800:DRAW 1279
,900:DRAW 0,900:DRAW 0,800
200 FORX=0TO1279 STEP 128:MOVEX,800:DR
AW X,900:NEXTX
210 VDU 5:GCOL 0,6
220 FORA=0 TO 49
230 Y=A DIV 10:X=A MOD 10
```



```

240     MOVE (X*128)-500,35+(Y*200):PRIN
TA
250     NEXTA
260 VDU 4:VDU 23;8202;0;0;0;0;
270 GCOL 0,1
280 MOVE 1100,800:DRAW 600,700
290 MOVE 50,800:DRAW 50,700
300 MOVE 974,600:DRAW 50,500
310 MOVE 178,400:DRAW 50,300
320 MOVE 562,400:DRAW 770,300
330 MOVE 400,200:DRAW 302,100
340 MOVE 1100,200:DRAW 600,100
350 GCOL 0,2
360 MOVE 50,100:DRAW 178,200
370 MOVE 462,100:DRAW 562,200
380 MOVE 302,300:DRAW 402,400
390 MOVE 1200,300:DRAW 690,400
400 MOVE 1100,500:DRAW 1100,600
410 MOVE 1200,700:DRAW 1200,800
420 C=0:P=0
430 GCOL 0,5:VDU 5
440 CX=C MOD 10: CX= CX*128+50: C1X= CX
450 CY=C DIV 10: CY= CY*200+100: C1Y= CY
460 MOVE CX,CY:PRINTCHR$(244)
470 PX=P MOD 10: PX= PX*128+50: P1X= PX
480 PY=P DIV 10: PY= PY*200+100: P1Y= PY
490 VDU 5:GCOL 0,4:MOVE PX,PY:PRINTCHR
$(245):VDU 4
500 VDU 4:VDU 23;8202;0;0;0;0;
510 R=RND(6)
520 PRINTTAB(0,1);"I roll ";R;"      "
530 C=C+R:FORH=1TO1000:NEXTH
540 RESTORE
550 FORA=1 TO 13
560     READF,G: IF C=F THEN C=G
570     NEXTA
580 VDU 5:GCOL 0,0:MOVE C1X,C1Y:PRINTC
HR$(244):VDU 4
590 CX=C MOD 10: CX= CX*128+50

```

```

600 CY=C DIV 10:CY=CY*200+100
610 VDU 5:GCOL 0,5
620 MOVE CX,CY:PRINTCHR$(244):VDU 4
630 PRINTTAB(0,1);"Press SPACE to roll
dice."
640 REPEAT R=RND(6):UNTIL INKEY$(0)="
"
650 PRINTTAB(0,1);"You roll ";R;"
"
660 PRINTTAB(0,2);"
"
670 FORH=1TO1000:NEXT
680 P=P+R
690 RESTORE
700 FORB=1TO13
710 I=0
720 READF,G:IF P=F THEN P=G:I=1
730 IF I=1 AND F<G THEN FORA=0 TO 2
00STEP5:SOUND 1,-15,A,1:NEXT
740 IF I=1 AND F>G THEN FORA=200 TO
0STEP-5:SOUND 1,-15,A,1:NEXT
750 NEXTB
760 VDU 5:GCOL 0,0:MOVE P1X,P1Y:PRINTC
HR$(245):VDU 4
770 PX=P MOD 10:PX=PX*128+50
780 PY=P DIV 10:PY=PY*200+100
790 VDU 5:GCOL 0,4
800 MOVE PX,PY:PRINTCHR$(245):VDU 4
810 SOUND 1,-15,P*4,5:FORH=1TO1000:NEX
TH
820 IF C>49THENPROC_CPW
830 IF P>49THENPROC_YW
840 GOTO430
850 DATA 0,11,3,14,13,2,18,4,12,23,19,
25,21,10,24,16,28,38,37,20,39,49,48,35,4
0,30
860 DEFPROC_CPW
870 GCOL 0,5:COLOUR 133
880 FORX=0TO1279STEP 4
890 MOVE X,0:DRAW X,1279

```

```

900    NEXTX
910 PRINTTAB(0,10);"Computer wins"
920 PRINT"Another go?";
930 A$=GET$:IF A$=" " THEN 930
940 PRINTA$
950 FORA=1TO100:NEXT
960 IFA$="Y" OR A$="y" THEN RUN
970 END
980 DEF PROC_YW
990 GCOL 0,4:COLOUR 132
1000 FORX=0TO1279STEP4
1010    MOVE X,0:DRAW X,1279
1020    NEXTX
1030 PRINTTAB(0,10);"You win"
1040 GOTO920

```

>

THE WELL

```

    0 REM THE WELL
  10 MODE 7
  20 PROCIN
  30 PROCVA
  40 MODE 5
  50 PROCDW
  60 PROCTW
  70 PROCD
  80 U=TIME
  90 REPEAT
100   PROCDS
110   UNTIL LZ>512 OR LZ<0
120 U=TIME-U
130 M%=(U DIV 6000)MOD 60
140 S%=(U DIV 100)MOD 60
150 IF LZ<0 THEN PROCKP ELSE PROCKH
160 PROCNG
170 END
180 END
190 DEF PROCIN
200 VDU 31,10,10,141
210 PRINT"The Well"
220 VDU 31,10,11,141
230 PRINT"The Well"
240 T=TIME
250 REPEAT
260   UNTIL INKEY(-99) OR TIME-T>200
270 ENDPROC
280 DEF PROCVA
290 LZ=400
300 DIM F$(1,4)
310 FOR A%=1 TO 4
320   READ F$(0,A%)
330   READ F$(1,A%)
340   NEXT
```

350 VDU 23,255,153,90,60,24,24,24,36,1
95

360 DATA+,plus,-,minus,*,times

370 DATA/,divided by

380 ENDPROC

390 DEF PROCDW

400 VDU 19,0,1,0,0,0

410 VDU 19,1,4,0,0,0

420 MOVE 128,895

430 DRAW 128,128

440 DRAW 512,128

450 DRAW 512,895

460 GCOL 0,1

470 MOVE 136,512

480 MOVE 504,512

490 PLOT 85,136,132

500 MOVE 504,132

510 PLOT 85,504,512

520 DEF PROCBK

530 GCOL 3,3

540 M%=1023-L%

550 MOVE 320,1023

560 DRAW 320,M%

570 GCOL 3,2

580 DRAW 320,M%-64

590 MOVE 272,M%

600 MOVE 312,M%

610 PLOT 85,312,M%-64

620 MOVE 328,M%

630 MOVE 368,M%

640 PLOT 85,328,M%-64

650 ENDPROC

660 DEF PROCTW

670 VDU 28,9,27,18,4

680 ENDPROC

690 DEF PROCD

700 PRINT"Please"

710 PRINT"Enter"

720 PRINT"Difficulty"

```

730 PRINT "1 Easy"
740 PRINT "20 Nasty"
750 COLOUR 1
760 INPUT 'D%
770 COLOUR 3
780 IF D%<1 OR D%>20 THEN PRINT "Must
be""between""1 and 20":GOTO 750
790 PRINT "What is""your name?"
800 COLOUR 1
810 INPUT 'A$
820 IF LENA$<1 OR LENA$>10 THEN 810
830 COLOUR 3
840 PRINT "Thankyou" 'A$
850 PROCT(100)
860 CLS
870 ENDPROC
880 DEF PROCDS
890 F%=RND(4)
900 IF F%>D% THEN 890
910 ON F% GOSUB 1710,1750,1800,1840
920 W%=EVAL(STR%X%+F$(0,F%)+STR$Y%)
930 PROCQ
940 ENDPROC
950 DEF PROCQ
960 Q$=STR%X%+CHR$10+CHR$10+CHR$13+F$(
1,F%)+CHR$10+CHR$10+CHR$13+STR$Y%+CHR$10
+CHR$13
970 PRINT "The mad"
980 PRINT "professor"
990 PRINT "says -"
1000 COLOUR 2
1010 PRINT 'A$
1020 PRINT "what is"
1030 *FX15,0
1040 PRINT 'Q$
1050 COLOUR 1
1060 T=TIME
1070 INPUT G%
1080 T=INT((TIME-T)/100+.5)

```

```

1090 COLOUR 3
1100 IF G%=W% THEN PRINT "You took";T'
"seconds" ELSE GOTO 1120
1110 PROCT(150)
1120 CLS
1130 IF W%<>G% THEN PRINT "You got it"
"wrong!!"
1140 PROCBK
1150 IF W%=G% AND T<30 THEN L%=L%-(32-T
)
1160 IF W%<>G% THEN L%=L%+20
1170 PRINT 'Q$
1180 PRINT "'equals "'W%
1190 PROCBK
1200 S=TIME
1210 REPEAT
1220 UNTIL TIME-S>500 OR INKEY(-99)
1230 CLS
1240 ENDPROC
1250 DEF PROCKH
1260 VDU 5
1270 VDU 19,2,2,0,0,0
1280 COLOUR 2
1290 MOVE 140,615
1300 PRINT "glug"
1310 PROCT(50)
1320 MOVE 208,581
1330 PRINT "glug"
1340 PROCT(50)
1350 MOVE 272,544
1360 PRINT "glug"
1370 PROCT(150)
1380 VDU 4
1390 CLS
1400 COLOUR 3
1410 PRINT "Bad luck""the mad""profess
or"
1420 PRINT "got you,"'A$
1430 PROCT(250)

```

```

1440 ENDPROC
1450 DEF PROCKP
1460 VDU 5
1470 GCOL 3,3
1480 A%=995
1490 REPEAT
1500     MOVE 200,A%
1510     VDU 255
1520     PROCT(5-10*(A%<512))
1530     SOUND 1,-10,A%/4,1-2*(A%<512)
1540     MOVE 200,A%
1550     VDU 255
1560     A%=A%-32
1570     UNTIL A%=163
1580 MOVE 200,A%
1590 VDU 255
1600 COLOUR 3
1610 VDU 4
1620 CLS
1630 PRINT"Well done."
1640 PRINT"You threw"
1650 PRINT"the mad"
1660 PRINT"professor"
1670 PRINT"into the"
1680 PRINT"well!!!"
1690 PROCT(250)
1700 ENDPROC
1710 REM +
1720 X%=RND(D%^2*10)
1730 Y%=RND(D%^2*10)
1740 RETURN
1750 REM -
1760 X%=RND(D%^2*10)+10
1770 Y%=RND(D%^2*10)
1780 IF Y%>X% THEN 1770
1790 RETURN
1800 REM x
1810 X%=RND(D%)
1820 Y%=RND(D%)

```



```

1830 RETURN
1840 REM /
1850 W%=RND(D%)
1860 Y%=RND(D%)
1870 X%=Y%*W%
1880 RETURN
1890 DEF PROCT(K%)
1900 J=TIME
1910 REPEAT UNTIL TIME-J>K%
1920 ENDPROC
1930 DEF PROCNG
1940 PRINT""You took "";M%;" minutes "
1950 PRINT ;S%;" seconds""
1960 PRINT"Do you""want""another""go
"
1970 PRINTA$
1980 PRINT"?"
1990 PRINT""Press"
2000 PRINT"Y or N"
2010 REPEAT
2020   UNTIL INKEY(-69) OR INKEY(-86)
2030 *FX15,0
2040 IF INKEY(-69) THEN RUN
2050 CLS
2060 ENDPROC

```

>

YAHM

```
0 REM YAHM
10 MODE 7
20 PROCIN
30 PROCVA
40 FOR A%=1 TO 13
50   PROC SH
60   PROC GO
70   PROC H
80   PROC PO
90   NEXT A%
100 PROC SH
110 FOR A%=1 TO 17
120   VDU 127
130   NEXT
140 PRINT "Play again? (Press Y or N)";
150 REPEAT
160   UNTIL INKEY(-69) OR INKEY(-86)
170 IF INKEY(-69) THEN RUN
180 *FX15,0
190 CLS
200 END
210 DEF PROCIN
220 VDU 31,10,10,141
230 PRINT "Yahm"
240 VDU 31,10,11,141
250 PRINT "Yahm"
260 T=TIME
270 REPEAT
280   A$=INKEY$(0)
290   UNTIL A$=" " OR TIME-T>200
300 ENDPROC
310 DEF PROCVA
320 DIM U(9)
330 DIM L(10)
340 DIM U$(9)
350 DIM L$(9)
```

```

360 RESTORE 510
370 FOR U=1 TO 9
380   READ U$(U),C
390   U$(U)=FNST(U$(U))
400   NEXT
410 FOR L=1 TO 9
420   READ L$(L),C
430   L$(L)=FNST(L$(L))
440   NEXT
450 DIM E(6)
460 E(0)=6
470 DIM D(5)
480 D(0)=5
490 DIM R(5)
500 R(0)=5
510 DATA Aces,129,Twos,131,Threes,130
520 DATA Fours,134,Fives,132,Sixes,133
530 DATA TOTAL,135,Bonus (if 63+),135
540 DATA Total of UPPER HALF,135
550 DATA 3 of a kind,129
560 DATA 4 of a kind,131
570 DATA Full House,130
580 DATA Small Straight,134
590 DATA Long Straight,132,YAHM,136
600 DATA Chance,133
610 DATA Total of LOWER HALF,135
620 DATA Total of UPPER HALF,135
630 ENDPROC
640 DEF PROC SH
650 CLS
660 VDU 131,157,132
670 PRINT "YAHM SCORE CARD"
680 VDU 134,157,132
690 PRINT STRING$(36,"_")
700 FOR U=1 TO 9
710   PRINT U$(U);TAB(25);-U(U)*(U(U)<
>-1)
720   NEXT
730 PRINT

```

```

740 FOR L=1 TO 9
750   PRINT L$(L);TAB(25);-L(L)*(L(L)<
>-1)
760   NEXT
770 PRINT
780 FOR A=0 TO 1
790   VDU 131+3*A,141
800   PRINT "Grand Total";TAB(23);L(10
)
810   NEXT
820 VDU 135,157,136,129
830 PRINT "hit the space-bar";
840 REPEAT A=GET
850   UNTIL A=32
860 ENDPROC
870
880
890 REM D=CHECK NO. (1-6 UH; 7-13 LH)
900 REM E=NO.POINTS SCORED THIS ROUND
910
920 REM 1-6 ACES-SIXES
930 FOR A=1 TO 5
940   E=E-(D(A)=D)
950   NEXT
960 E=E*D
970 IF E=0 THEN E=-1
980 RETURN
990
1000 REM 7/8 - 3/4 OF A KIND
1010 H1=0
1020 FOR A=1 TO 6
1030   H=0
1040   FOR B=1 TO 5
1050     H=H-(D(B)=A)
1060   NEXT
1070   IF H1<H THEN H1=H:H2=A
1080 NEXT
1090 IF D-H1>4 THEN E=-1 ELSE E=FNT0(E)
1100 RETURN
1110

```

```

1120 REM 9 - FULL HOUSE
1130 C1=0
1140 C2=0
1150 FOR A=1 TO 5
1160     IF C1=0 THEN C1=D(A)
1170     IF D(A)<>C1 AND C2=0 THEN C2=D(A
)
1180     IF D(A)<>C1 AND D(A)<>C2 THEN E=
-1
1190     NEXT
1200 IF C2=0 THEN E=25:RETURN
1210 IF E=-1 THEN RETURN
1220 T1=0
1230 T2=0
1240 FOR A=1 TO 5
1250     T1=T1-(D(A)=C1)
1260     T2=T2-(D(A)=C2)
1270     NEXT
1280 IF T1>3 OR T2>3 THEN E=-1 ELSE E=2
5
1290 RETURN
1300
1310 REM 10/11 - LOW/HIGH STRAIGHT
1320 FOR A=1 TO 5
1330     E(D(A))=D(A)
1340     NEXT
1350 T=0
1360 FOR A=1 TO 5
1370     T=T-(E(A)=0)
1380     NEXT
1390 E=(T>2)+(T>1)*(D=11)+(E(1)<>0)*(E(
6)<>0)*(T=1)
1400 IF E=-1 THEN RETURN
1410 IF T=1 AND D=11 THEN E=40:RETURN
1420 IF T=1 AND D=10 THEN E=30:RETURN
1430 G=1
1440 GP=0
1450 FOR A=1 TO 6
1460     GP=(E(A)=0)

```

```

1470 IF GP=-1 THEN G=G+1
1480 IF E(A)=0 THEN GP=0
1490 NEXT
1500 IF G>2 AND G<6 THEN E=-1:RETURN
1510 E=30
1520 RETURN
1530
1540 REM 12 - YAHTZEE
1550 T=D(1)
1560 FOR A=2 TO 5
1570 IF D(A)<>T THEN E=-1
1580 NEXT
1590 IF E<>-1 THEN E=50
1600 RETURN
1610
1620 REM 13 - CHANCE
1630 E=FNTD(E)
1640 RETURN
1650 DEF FNTD(X)
1660 FOR A=1 TO 5
1670 X=X+D(A)
1680 NEXT
1690 =X
1700 DEF PROC60
1710 A=0
1720 PROC1D
1730 FOR AG=3 TO 1 STEP -1
1740 CLS
1750 PRINT "YOU HAVE ";AG;" THROWS"
1760 PRINT TAB(30,22);"shaking"
1770 PROCSC
1780 FOR B=1 TO 10
1790 FOR C=1 TO 5
1800 IF R(C)=0 THEN D(C)=RND(6)
1810 PRINT TAB(C*5,22);D(C)
1820 NEXT
1830 NEXT
1840 PRINT TAB(30,22);" "
1850 FOR A=1 TO 5

```

```

1860      R(A)=1
1870      NEXT
1880      IF AG=1 THEN ENDPROC
1890      PRINT TAB(0,15);"Which dice to r
eshake? (press 115)"
1900      PRINT TAB(0,18);"Space-bar to th
row dice"
1910      REPEAT
1920          A=GET
1930          UNTIL A>48 AND A<54 OR A=32
1940          IF A=32 THEN NEXT
1950      A=A-48
1960      VDU 31,0,23,129,31,A*5,23,94+(R(
A)=0)*62
1970      IF R(A)=1 THEN R(A)=0 ELSE R(A)=1
1980      GOTO 1910
1990      DEF PROCSC
2000      PRINT TAB(0,0);
2010      FOR U=1 TO 6
2020          IF U(U)=0 THEN PRINTU$(U);TAB(25
);U
2030      NEXT
2040      PRINT
2050      FOR L=1 TO 7
2060          IF L(L)=0 THEN PRINTL$(L);TAB(25
);L+6
2070      NEXT
2080      PRINT'''
2090      ENDPROC
2100      DEF PROCH
2110      PROCSC
2120      VDU 31,0,15,131,157,136,132
2130      PRINT "Which option ???"'
2140      GOSUB1550
2150      IF E=50 AND L(6)>0 THEN L(6)=L(6)
+100
2160      INPUT "Enter one of above numbers
"D
2170      E=0

```

```

2180 IF D>6 THEN 2210
2190 IF D<7 AND (D<1 OR U(D)<>0) THEN 2
120
2200 IF D<7 THEN 2220
2210 IF D>6 AND (D>13 OR L(D-6)<>0) THE
N 2120
2220 E=0
2230 IF D<7 THEN GOSUB 890:ENDPROC
2240 ON D-6 GOSUB 1010,1010,1130,1310,1
310,1550,1630
2250 ENDPROC
2260 DEF PROCPO
2270 IF D<7 THEN U(D)=E
2280 IF D>6 THEN L(D-6)=E
2290 U(7)=0
2300 FOR A=1 TO 6
2310   U(7)=U(7)-U(A)*(U(A)<>-1)
2320   NEXT
2330 U(8)=- (U(7)>62)*35
2340 U(9)=U(7)+U(8)
2350 L(8)=0
2360 FOR A=1 TO 7
2370   L(8)=L(8)-L(A)*(L(A)<>-1)
2380   NEXT
2390 L(9)=U(9)
2400 L(10)=L(8)+L(9)
2410 ENDPROC
2420 DEF FNST(X$)
2430 X$=CHR$(C)+X$
2440 X$=X$+STRING$(25-LEN(X$),".")
2450 =X$
2460 DEF PROCCLD
2470 FOR A=1 TO 5
2480   D(A)=0
2490   R(A)=0
2500   E(A)=0
2510   NEXT
2520 ENDPROC

```

>

APPENDIX A

Best Scores

Here is a list of the best scores, to give you something to aim for when playing the games. If you beat any of them (which we are sure you will) write to us and we will try to include your score in any future editions.

18th hole	Hole in one
3D Invaders	12,925
Aspirin	33.87 (stomach 1)
Balloons	779
Battlefield	Beat computer 12–8
Big Business	£3,476
Breakout	1,970
'Copter	3,308 (speed 1)
Defender	4,800 (4pc, 3in)
DR 2000	1,150
Dodge	8.76 (level 3)
Downhill Skiing	2:05,26 (71 crashes)
Fruit Machine	£400
Guess	Beat computer 15–21
Hop It!	490
Lander	4,534 fuel left
Invaders	1,520
Nightfall	9,119
Portals	315
Rat Attack	890
Shock	10,950
Slaves	922% profit
Target	7 (speed 5)

APPENDIX B

Joysticks

If you are lucky enough to own a pair of joysticks, you may find the following information useful when converting the games.

3D Invaders

```
160 IF (ADVAL0 AND 3) = 1 THEN PROCJJ:PROCLL
1010 IF (ADVAL0 AND 3) = 1 THEN PROCJJ:GOTO 1190
1660 DEF PROCJJ
1670 REPEAT
1680 UNTIL (ADVAL0 AND 3) < > 1
1690 ENDPROC
1410 U% = U% + ((ADVAL1 > 44000) - (ADVAL1 <
      21000)) * 16
```

Animate

```
240 X = X + (ADVAL1 > 44000) - (ADVAL1 < 21000)
250 Y = Y + (ADVAL2 < 21000) - (ADVAL2 > 44000)
```

Balloons

```
190 U% = U% + ((ADVAL1 > 44000) - (ADVAL1 <
      21000)) * 32
500 REPEAT:UNTIL (ADVAL0 AND 3) = 1
```

Breakout

```
490 B = B + (ADVAL1 > 44000) - (ADVAL1 < 21000)
```

'Copter

```
300  U% = U% + ((ADVAL1 > 44000) - (ADVAL1 < 21000)) * 16
310  P% = P% - ((ADVAL0 AND 3) = 1) * 32
360  S% = S% + ((ADVAL0 AND 3) = 1) * 35
1120 UNTIL (ADVAL0 AND 3) = 1
```

Downhill Skiing

```
210  X1 = X1 + (ADVAL1 > 44000) - (ADVAL1 < 21000)
```

Invaders

```
440  IF (ADVAL1 > 44000) THEN PX = PX - 0.1
450  IF (ADVAL1 < 21000) THEN PX = PX + 0.1
470  IF (ADVAL0 AND 3) = 1 THEN PROCFIRE
```

Nightfall

```
150  IF (ADVAL0 AND 3) = 1 THEN PROCB
```

Parachute

```
230  W = W + (ADVAL1 > 44000) - (ADVAL1 < 21000)
```

Portals

```
310  G% = G% + ((ADVAL1 < 21000) - (ADVAL1 > 44000)) * P
```

(This is correct)

Rat Attack

Delete line 390

```
400  IF ADVAL1 > 44000 AND PX > 1 THEN MAN$ = CHR$244:PX = PX - 1
410  IF ADVAL1 < 21000 AND PX < 17 THEN MAN$ = CHR$242:PX = PX + 1
420  IF (ADVAL0 AND 3) = 1 AND MAN$ < > CHR$245 AND MAN$ < > CHR$243 THEN MAN$ = CHR$(ASC(MAN$) + 1):PROCSCAN:ENDPROC
```

```

430 IF (ADVAL0 AND 3)=1 AND MAN$<>CHR$244
    AND MAN$<>242 THEN MAN$=CHR$(ASC
    (MAN$)-1):PROCSCAN

```

Shark

```

390 IF ADVAL1>44000 THEN PX=PX-1
400 IF ADVAL1<21000 THEN PX=PX+1
410 IF ADVAL2<21000 THEN PY=PY+1
420 IF ADVAL2>44000 THEN PY=PY-1

```

Shock

```

150 IF (ADVAL0 AND 3)=1 THEN PROCSB
980 U%=U%+((ADVAL1>44000)-(ADVAL1<2
    2000))*16

```

Smash Chamber

```

940 D%=- (ADVAL2>44000)-(ADVAL1<21000)*2
    -(ADVAL2<21000)*3-(ADVAL1>44000)*4

```

Target

```

310 IF (ADVAL0 AND 3)=1 THEN PROC_FIRE
570 X%=X%+(ADVAL1>44000)-(ADVAL1<2
    1000)
400 X%=X%+(ADVAL1>44000)-(ADVAL2<2
    1000)

```

APPENDIX C

Morse Code

A	J	S
.-	.-.-	...-
B	K	T (3 units)
-....	-.-	-
C	L	U
-.-.	.-..	..-
D	M	V
-..	--	...-
E (1 unit)	N	W
.	-.	.-
F	O	X
....	---	-.-
G	P	Y
--.	.-.	-.-
H	Q	Z
....	--.	---.
I	R	
..	.-.	
1	5	9
.----		-----
2	6	zero
..----	-.....	-----
3	7	
....--	--....	
4	8	
.....	----.	

APPENDIX D

Attributions

Tim D Rogers:

18th Hole; 3D Invaders; Aspirin; Balloons; Copter; Guess; Lander; Nightfall; Poem; Portals; Shock; Slaves; Smash Chamber; The Well; Yahm

Chris Callender:

Animate; Battlefield; Big Business; Breakout; The Clever Computer; Dodge; Downhill Skiing; Fruit Machine; Gold Mine; Hop It!; Morse Test; Parachute; Pilot; Shark; Synthesiser; Target; Ups 'n' Downs

Grant Evans:

Defender; DR 2000; Invaders; Rat Attack

Something tells me your BBC Micro is never going to recover after you've subjected it to a few rounds of these fast-moving games.

Tim D Rogers and Chris Callender have produced 36 outstanding games for the BBC Micro, 36 games you'll want to play again and again.

The book includes:

- 3D INVADERS
- PORTALS
- SYNTHESIZER
- BIG BUSINESS
- BREAKOUT
- PILOT
- GOLDMINE
- HOP IT!

From graphic adventure programs to fast-moving arcade action, this book gives you a whole library of software-standard games.

**Another great book from
Interface Publications**

Part of Tim Hartnell's *Success in the Fast Lane* computer series